SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Car Damage Detection Using Computer Vision

SUBMITTED TOWARDS THE
FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Atharva Kasar	B191054257
Apurva Kshirsagar	B191054268
Geeta Hade	B191054242
Nishant Khandhar	B191054261

Under The Guidance of

Prof. P.B. Warungse

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

Sinhgad Technical Education Society's

NBN Sinhgad Technical Institutes Campus

NBN Sinhgad School of Engineering, Ambegaon (Bk),Pune-411041

Savitribai Phule Pune University

Academic Year: 2023-24

Sinhgad Institutes

Sinhgad Technical Education Society's

NBN Sinhgad Technical Institutes Campus

NBN Sinhgad School of Engineering, Ambegaon (Bk),Pune-411041

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

Car Damage Detection using Computer Vision

Submitted by

Atharva Kasar	B191054257
Apurva Kshirsagar	B191054268
Geeta Hade	B191054242
Nishant Khandhar	B191054261

is a bonafide work carried out by Students under the supervision of Prof. P. B. Warungse and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering) Project.

Prof. Priti B. Warungse
Internal Guide
Dept. of Computer Engg.

Dr. S. P. Bendale
H.O.D
Dept. of Computer Engg.

Dr. S. P. Patil
Principal
NBN Sinhgad Technical Institute Campus

Signature of Internal Examiner

Signature of External Examiner

PROJECT APPROVAL SHEET

A Project Title

Car Damage Detection using Computer Vision

Is successfully completed by

Atharva Kasar	B191054257
Apurva Kshirsagar	B191054268
Geeta Hade	B191054242
Nishant Khandhar	B191054261

at

DEPARTMENT OF COMPUTER ENGINEERING

Sinhgad Technical Education Society's

NBN Sinhgad Technical Institutes Campus

NBN Sinhgad School of Engineering, Ambegaon (Bk),Pune-411041

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2023-24

Prof. Priti B. Warungse
Internal Guide
Dept. of Computer Engg.

Dr. Shailesh P. Bendale
H.O.D
Dept. of Computer Engg.

Abstract

This research paper introduces an extensive framework for the detection of car damage through the application of deep learning methodologies. The proposed system seeks to fulfill the crucial demand for accurate, efficient, and automated techniques in assessing vehicle damage, offering potential applications in insurance claims processing, vehicle maintenance, and accident analysis. In our devised solution, we have integrated two convolutional neural network (CNN) models to address the task at hand. VGG16 is employed for the identification of car damage, determining the damage's location, and assessing its severity. Concurrently, Mask RCNN is utilized to precisely mask out the damaged region. The combined output from both models provides a comprehensive understanding of the damage inflicted on the car. The deep learning model undergoes training on a meticulously curated dataset of annotated vehicle images, highlighting the presence and severity of damage. Rigorous testing and validation procedures are employed to evaluate the system's accuracy, precision, recall, and F1-score. The resultant system showcases the capacity to significantly streamline the processes of car damage assessment, diminish human error, and expedite the settlement of insurance claims.

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on ***'Car Damage Detection using Computer Vision'****.*

I would like to take this opportunity to thank my internal guide ***Prof. P.B. Warungse*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.*

I am also grateful to ***Dr. S.P. Bendale****, Head of Computer Engineering Department, CollegeName for his indispensable support, suggestions.*

In the end our special thanks to ***Prof. P.B. Warungse and Prof. S.C. Sethi*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Project.*

Atharva Kasar
Apurva Kshirsagar
Geeta Hade
Nishant Khandhar
(B.E. Computer Engg.)

INDEX

List of Figures

List of Tables

CHAPTER 1

SYNOPSIS

1.1 PROJECT TITLE

Car Damage Detection Using Computer Vision

1.2 PROJECT OPTION

Internal Project

1.3 INTERNAL GUIDE

Prof. P. B. Warungse

1.4 SPONSORSHIP AND EXTERNAL GUIDE

Not Applicable

1.5 TECHNICAL KEYWORDS (AS PER ACM KEYWORDS)

1. Car Damage Detection
2. Convolutional Neural Networks (CNNs)
3. Supervised Learning
4. Vehicle Damage Assessment
5. Deep Learning
6. Image Processing

1.6 PROBLEM STATEMENT

In the automotive industry, timely and accurate assessment of vehicle damages is crucial for insurance claims and repair processes. However, the manual inspection of vehicles for damages is time-consuming, subjective, and often prone to errors. The challenge is to develop an automated car damage detection system using machine learning and computer vision techniques.

1.7 ABSTRACT

This research paper presents a comprehensive framework for car damage detection using deep learning techniques. The proposed system aims to address the critical need for accurate, efficient, and automated methods for assessing vehicle damage, with potential applications in insurance claims processing, vehicle maintenance, and accident analysis.The project leverages a state-of-the-art convolutional neural network (CNN) architecture, specifically ResNet, for its superior feature extraction capabilities and classification performance. The deep learning model is trained on a carefully curated dataset of vehicle images, annotated with labels indicating the presence and severity of damage. The project undergoes rigorous testing and validation to assess its accuracy, precision, recall, and F1-score. User feedback and user experience evaluations are considered for continuous improvement. The resulting system demonstrates the potential to significantly streamline car damage assessment processes, reduce human error, and expedite insurance claim settlements. The project undergoes rigorous testing and validation to assess its accuracy, precision, recall, and F1-score

1.8 GOALS AND OBJECTIVES

- Gather and annotate a diverse dataset of vehicle images for training and validation.
- Develop a robust machine learning model tailored for accurate detection of vehicle damages.
- Integrate advanced computer vision techniques for efficient image analysis.
- Optimize the model to facilitate real-time processing and immediate damage assessment.
- Continuously fine-tune the model and incorporate feedback loops to enhance accuracy.
- Seamlessly integrate the system with existing processes and workflows.
- Design an intuitive user interface for easy interaction and interpretation of results.

1.9 RELEVANT MATHEMATICS ASSOCIATED WITH THE PROJECT

System Description:

- Input: Images of damaged vehicles for analysis.
- Output: Detected and classified car damages (e.g., dents, scratches) with their locations and severity.
- Data Structures: Image representation using matrices (pixels and color channels). Lists or arrays to store and manipulate image data. Trees or graphs to represent hierarchical features.
- Functions : PreprocessImage(image): Preprocesses the input image for analysis (e.g., resizing, normalization). DetectDamage(image): Detects damage within the image using computer vision techniques. ClassifyDamage(damage): Classifies the detected damage based on severity and type.
- Mathematical formulation:

    1. Precision: Precision is the ratio of true positive predictions to the total positive predictions made by the model. It measures the accuracy of the positive predictions. Precision is calculated as:

    $$\text{Precision} = \frac{TruePositives}{TruePositives+FalsePositives}$$

    2. Recall (Sensitivity or True Positive Rate): Recall is the ratio of true positive predictions to the total actual positive instances in the dataset. It measures the model's ability to identify all relevant instances. Recall is calculated as:

    $$\text{Recall} = \frac{TruePositives}{TruePositives+FalseNegatives}$$

    3. F1 Score: The F1 score combines precision and recall into a single metric, providing a balance between them. It is calculated as the harmonic mean of precision and recall

    $$\text{F1 Score} = \frac{2.Precison.Recall}{Precision+Recall}$$

- Success Conditions: Accurate detection and classification of car damages. Efficient parallel processing with minimal time and resource usage. Reliable preprocessing leading to improved damage detection accuracy.
- Failure Conditions: Inaccurate detection or misclassification of car damages. Excessive processing time or resource utilization, causing delays.Preprocessing errors affecting damage detection results

1.10 NAMES OF CONFERENCES / JOURNALS WHERE PAPERS CAN BE PUBLISHED

- IEEE/ACM Conference/Journal 1
- IOSR Journal of Engineering (IOSR - JEN)
- International Journal of Information Technology - Sprinjer
- INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS - IJCRT

1.10.1 Sem-1 Paper:

1. Paper Title: Car Damage Detection Using Computer Vision
2. Name of the Conference/Journal where paper submitted : Journal of Emerging Technologies and Innovative Research - (JETIR)
3. Paper accepted/rejected : Published
4. Date Published : 17 Nov'23
5. Review comments by reviewer :NA
6. Corrective actions if any : NA

1.10.2 Sem-2 Paper:

1. Paper Title: Car Damage Detection Using Computer Vision
2. Name of the Conference/Journal where paper submitted : Microsoft CMT (NCCC)
3. Date Submitted : 5 April'24
4. Paper accepted/rejected : In-Process
5. Review comments by reviewer : NA
6. Corrective actions if any : NA

1.11 REVIEW OF CONFERENCE/JOURNAL PAPERS SUPPORTING PROJECT IDEA

[1] M. Chen, F. Bai and Z. Gerile, Special Object Detection Based On Mask Rcnn, 2021 17th International Conference on Computational Intelligence and Security (CIS), 2021, pp. 128-132, doi: 10.1109/CIS54983.2021.00035

[2] M. Ye et al., A Lightweight Model of VGG-16 for Remote Sensing Image Classification, in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 6916-6922, 2021, doi: 10.1109/JSTARS.2021.3090085.

[3] 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), 2021, pp. 1-5, doi: 10. 1109/ICCICT50803.2021.9510069.

[4] P. M. Kyu and K. Woraratpanya, Car Damage Detection and Classification, in Proceedings of the 11th International Conference on Advances in Information Technology, Bangkok Thailand, Jul. 2020, pp. 1-6, doi:10.1109

[5] H. Patel, hemilpate1971/Damage-car-detection, GitHub, Aug 13, 2019. (accessed Feb. 10, 2021).

1.12 PLAN OF PROJECT EXECUTION

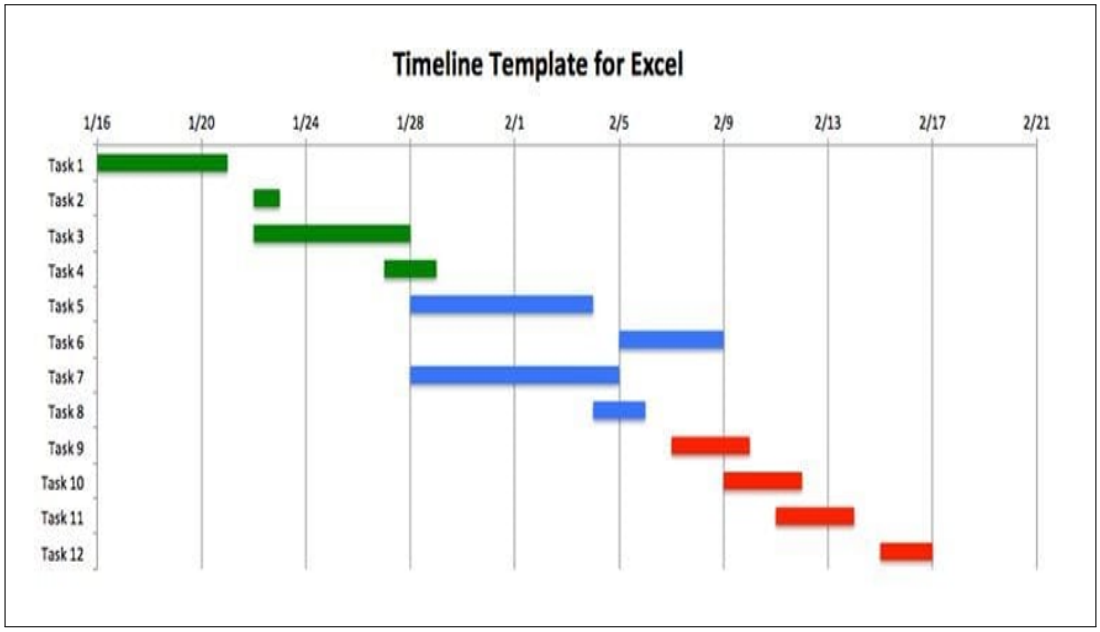


Figure 1.1: Project Plan

CHAPTER 2

TECHNICAL KEYWORDS

2.1 AREA OF PROJECT

Computer Vision, Image Processing, Machine Learning

2.2 TECHNICAL KEYWORDS

1. Computer Vision
2. Image Processing
3. Machine Learning (ML)
4. Deep Learning (DP)
5. Convolutional Neural Network (CNN)
6. Feature Extraction
7. Object Detection
8. Region Of Interest (ROI)
9. Anomaly Detection)
10. Segmentation
11. Paging
12. Classification
13. Accuracy
14. Precision
15. Recall
16. F1-Score
17. Training and Testing Dataset
18. ROI Localization

19. Real-Time Detection
20. Transfer Learning
21. Data Augumentation

CHAPTER 3

INTRODUCTION

3.1 PROJECT IDEA

Create an automated Car Damage Detection System that utilizes computer vision and machine learning to analyze images of cars for various types of damage, such as dents, scratches, and broken lights. The system will include data collection and annotation, model training, a user-friendly interface for damage detection and severity assessment, reporting, and integration options for user databases. It aims to streamline damage assessment for insurance companies, car repair shops, and vehicle owners, offering a comprehensive solution for detecting, categorizing, and reporting car damage, ultimately improving efficiency and objectivity in repair recommendations.

3.2 MOTIVATION OF THE PROJECT

Car damage detection is motivated by the need for a more efficient, objective, and technology-driven approach to assessing vehicle condition. This system addresses the inconvenience and subjectivity in evaluating car damage, benefiting insurance companies, car repair services, and vehicle owners alike. By automating the process through computer vision and machine learning, it reduces human error, speeds up insurance claims processing, aids in repair shop recommendations, and empowers individuals to make informed decisions about vehicle maintenance and repair, ultimately improving safety and convenience within the automotive industry.

Car damage detection is motivated by the desire to enhance safety, convenience, and cost-efficiency in the automotive industry. Accurate and swift identification of car damage is crucial for insurance claims, timely repairs, and ensuring roadworthiness. Automation through computer vision and machine learning technologies offers a more objective and scalable approach, reducing human error and subjective judgment. This technology-driven solution streamlines the assessment process, facilitates quicker insurance claim settlements, enables proactive maintenance, and contributes to safer road conditions by ensuring damaged vehicles are promptlyrepaired.

3.3 LITERATURE SURVEY

There are many studies conducted for the car damage detec- tion. Majority among them are using one of these pre- trained models for feature extraction and classification. References

[1] **Title:** Automatic vehicle damage detection classification framework using fast and mask deep learning

Author: D. A. Reddy, Saroj sambharkar, Chandrashekhar bhoyar

Year: 2022

Description: In this study, Reddy, Sambharkar, and Bhoyar proposed a framework aimed at predicting the type of vehicle damage, whether it's minor or major. The framework leverages machine learning algorithms, tapping into the advancements in artificially intelligent systems. The proposed system offers a promising approach to automating the identification and classification of vehicle damage, contributing to the efficiency and accuracy of damage assessmentprocesses.

[2] **Title:** Car Damage Identification and Categorization Using Various Transfer Learning Models

Author: Sruthy C. M., Nandakumar R.

Year: 2021

Description: Sruthy C. M. and Nandakumar R. conducted a study on car damage identification and categorization using convolutional neural networks (CNNs) with various transfer learning models. The research explores the application of deep learning techniques for automating car damage recognition and classification tasks. The study specifically focuses on keywords such as "bumper dent," "door dent," "glass shatter," etc., highlighting different types of car damage that can be detected and categorized using machine learningapproaches.

[3] **Title:** Car damage detection and assessment using CNN
Author: Atharva Shirode, Tejas Rathod, Parth Wanjari, Aparna Halbe
Year: 2021
Description: Shirode, Rathod, Wanjari, and Halbe implemented a project in two parts. The first part involved training three models using transfer learning with VGG models. In the second part, Mask R-CNN was utilized for localizing car damage. This study presents an approach to car damage detection and assessment using convolutional neural networks (CNNs), showcasing the effectiveness of deep learning techniques in automating damage detectiontasks.

[4] **Title:** Assessing Car Damage with Convolutional Neural Networks
Author: H. Bandi, S. Joshi, S. Bhagat and A. Deshpande
Year: 2021
Description: This paper applies Convolutional Neural Networks (CNNs) to damaged car images to assess the extent of damage. The authors utilize transfer learning to evaluate the merits of object recognition models available for this task. The study aims to automate the assessment of car damage using deep learning techniques, contributing to improved efficiency and accuracy in damage evaluationprocesses.

[5] **Title:** Review of deep learning: concepts, CNN architectures, challenges, applications, future directions
Author: Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ye Duan
Year: 2021
Description: In this paper, an overview of Deep Learning (DL) is presented, covering various perspectives such as the main concepts, architectures, challenges, applications, computational tools, and evolution matrix. Convolutional Neural Network (CNN) is highlighted as one of the most popular and widely used DL networks, contributing to the popularity of DL in contemporary times. CNN's main advantage over its predecessors lies in its ability to automatically detect significant features without human supervision, making it the most utilized DLarchitecture.

[6] **Title:** A Lightweight Model of VGG-16 for Remote Sensing Image Classification

Author: M. Yeetal

Year: 2021

Description: This paper explores the application of deep learning-based algorithms, specifically VGG16 and VGG19, to the domain of car damage detection and evaluation using real-world datasets. The authors investigate the impact of domain-specific pre-trained CNN models initially trained on the ImageNet dataset and further fine-tuned in this study. They employ transfer learning techniques to enhance the accuracy of pre-trained VGG models, alongside other methodologies. By combining transfer learning and L2 regularization, they achieve high accuracy in damaged detection, damage localization, and damage severity assessment. Their findings suggest that VGG19 outperforms VGG16, achieving an accuracy of 95.22%.

[7] **Title:** Automated detection of multiclass vehicle exterior damages using deep learning

Author: Maleika khan, Mohammad Hussain

Year: 2020

Description: Khan and Hussain proposed a system that enables users to upload vehicle images via a developed mobile application. The study utilizes a pre-trained deep learning model to recognize and classify multiple types of vehicle damages from the uploaded images. This research contributes to automating the detection and categorization of vehicle exterior damages through the application of deep learning techniques, enhancing the efficiency of damage assessment processes.

[8] **Title:** Overview of the development of AI dataset annotation

Author: Bochao Ao, Bingbing Fan

Year: 2020

Description: This paper analyzes the development history of AI dataset annotation, providing insights into the evolution of annotation techniques. It summarizes the general framework of AI dataset annotation and discusses three semi-automatic or automatic annotation methods. The paper compares and analyzes the advantages and disadvantages of these methods, offering valuable insights into the annotation process in the context of artificialintelligence.

[9] **Title:** Automatic Car Damage Recognition using Convolutional Neural Networks

Author: Jeffrey de Deijn

Year: 2018

Description: In this research, convolutional neural networks (CNNs) are employed for the automatic recognition of car damage in images, determining whether a given car image exhibits any damage. The study showcases that their transfer learning approach based on VGG16 can yield promising results particularly when the dataset is extensive and of superior quality. They achieve accurate classification of damage type with 75.1% accuracy, damage location with 68.7% accuracy, and damage size with 54.2% accuracy.

[10] **Title:** Vehicle Damage Severity Estimation for Insurance Operations Using In-The-Wild Mobile Images

Author: Dimitrios Mallios, Li Xaofie, Niall Mclaughlin, Jesus Martinez, Clare Galbraith

Year: 2016

Description: In this work, the authors propose a pipeline that utilizes photographs of a damaged car, collected by users from multiple angles, along with structured data about the vehicle. The goal is to estimate damage severity following an accident. The proposed approach leverages in-the-wild mobile images to enhance the accuracy of damage severity estimation, which can be valuable for insuranceoperations.

CHAPTER 4

PROBLEM DEFINITION AND SCOPE

4.1 PROBLEM STATEMENT

The problem revolves around inefficient and subjective car damage assessment in insurance claims and repairs. Manual inspections are slow, prone to human error, and lack data-driven insights. The process can be complex for car owners and may not adapt to specific business needs or various car models.The project aims to streamline assessment, reduce errors, provide data-driven insights, offer user-friendliness, customization, and scalability.

4.1.1 Goals and objectives

Software Goals and Objectives:

- Goal: Efficient car damage assessment.
- Objectives: Accurate severity and location classification, data-driven decision support, user-friendliness, customization, scalability, and continuous improvement.

Input Description:

- Input: Damaged vehicle image (JPEG/PNG).
- Size: Typically 800x600 pixels.
- Validation: Format and size checks.
- Dependency: Image quality and clarity.

Output Description:

- Output: Severity (Minor/Moderate/Major) and location (Rear Side/Back Side/-Side View).
- Presentation: User-friendly interface.
- Optional: Custom reports and integration-ready data.

4.1.2 Statement of scope

The software is designed for efficient car damage assessment, without delving into implementation details. It accepts damaged vehicle images in JPEG or PNG format, typically around 800x600 pixels. Input validation includes checks for format and file size, ensuring that it handles images within a 5MB limit. The accuracy of the assessment is dependent on the quality and clarity of the input image. The software operates within a two-state model, transitioning from ”Input Image Processing” to ”Result Presentation” after successful processing. Major inputs are limited to the image itself, while major outputs encompass damage severity and location classification, alongside a user-friendly visual representation. Optionally, the software can generate custom reports and provide integration-ready data for external systems.

The car damage detection software is designed to assess vehicle damage severity and location based on uploaded images. It will accurately classify damages as Minor, Moderate, or Major and tag their specific location as Rear Side, Back Side, or Side View. The software will offer a user-friendly web interface for image upload and result presentation. It will validate input images for format and size, provide real-time error handling, and deliver responsive user experiences. The software’s scope includes the potential for generating customized assessment reports and integration with external systems. It will not provide specific repair instructions, conduct physical inspections, or offer real-time image capture functionality.

4.2 SOFTWARE CONTEXT

The software is designed for car damage assessment, serving the automotive insurance industry, repair services, and car owners. It streamlines the assessment of damage severity and location, enabling efficient claim processing, data-driven decision-making, and trend analysis.

4.3 MAJOR CONSTRAINTS

1. Image Quality: Accuracy depends on image quality.
2. Hardware and Bandwidth: Must work on diverse devices and connections.
3. Regulatory Compliance: Must adhere to data privacy laws.
4. Scalability: Should handle increased traffic and data.
5. Integration Compatibility: Compatible with various external systems.
6. Data Diversity: Requires diverse training data for deep learning.
7. Labeling Accuracy: Accurate labels are vital for assessment.

4.4 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

- Problem solving involves considering multiple approaches to address the same issue. Efficiency issues are central to this consideration, with performance parameters as key factors. It's vital to assess and choose the most efficient solution, often factoring in resource utilization, execution speed, scalability, and adaptability. The goal is to find a solution that optimizes resources and minimizes bottlenecks, ensuring effective problem resolution.

4.5 SCENARIO IN WHICH MULTI-CORE, EMBEDDED AND DISTRIBUTED COMPUTING USED

1. **Multi-Core Computing**: Image Processing: Multi-core processors are utilized to process and analyze a high volume of vehicle damage images simultaneously. Each core can handle a different image, significantly reducing the time required for assessment and improving real-time processing. Parallel Feature Extraction: Multi-core computing can parallelize feature extraction from images, enhancing the system's ability to identify and classify damage patterns efficiently.

2. **Embedded Computing**: Onboard Vehicle Systems: Embedded systems within vehicles can play a vital role in damage detection. They can process data from in-car cameras and sensors, enabling features like real-time damage alerts and collision avoidance. IoT-Enabled Damage Assessment: Embedded IoT devices within the vehicle can capture and process images of potential damage, immediately reporting the assessment results to the driver and external systems.

3. **Distributed Computing**: Cloud-Based Analysis: Distributed computing connects the local system to cloud servers for advanced image analysis. Cloud-based systems can process and compare damage images with a vast database of known damage patterns, improving accuracy. Scalable Assessment: Distributed computing allows for the integration of various car damage detection systems across a wide area, ensuring scalable assessment capabilities for a large fleet or multi-location operations.

4.6 OUTCOME

The project will deliver:

- Efficient and accurate car damage assessment.
- Data-driven insights for informed decisions.
- User-friendly accessibility.
- Customization and integration options.
- Continuous model improvement.
- Scalability across various car models, benefiting car owners, insurers, and repair services.

4.7 APPLICATIONS

- Automotive insurance claim assessment.
- Vehicle safety features and collision avoidance.
- Fleet management for damage monitoring.
- Real-time car damage alerts for drivers.
- Car maintenance and repair services optimization.

4.8 HARDWARE RESOURCES REQUIRED

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	2 GHz	Necessary for real-time image processing.
2	RAM	3 GB	Essential for image data storage.

Table 4.1: Hardware Requirements

4.9 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System: Linux or Windows
2. IDE: Visual Studio Code, Atom, Jupyter Notebook or PyCharm
3. Programming Language: Python 3.7 or higher

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATES

Hybrid Project Management: Some projects, including AI projects, use a hybrid approach that combines elements of both Agile and traditional project management (e.g., Waterfall). This approach might be suitable for car damage detection systems where certain project phases are well-suited to Waterfall (e.g., requirements and system design) while others benefit from Agile (e.g., model developmentandtesting).

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

The cost estimation in Indian Rupees (INR) can be adjusted to more affordable levels, considering that the project may have fewer resources and a smaller scope. Here's a revised cost estimation suitable for a college-level project in India:

- Data Collection and Annotation: 2,000 - 5,000
- Model Development 5,000 - 6,000
- User Interface Development: 2,000 - 5,000
- Integration and Testing: 4,000 - 5,000
- User Education and Documentation: 2,000 - 5,000
- Other Costs (server infrastructure, tools, licenses, etc.): 2,000 - 4,000
- Total estimated cost: 15,000 - 30,000

5.1.1.2 Time Estimates

The development timeline for a car damage detection system can range from a few months to a year or more, depending on the project's scope. Here's a rough breakdown:

- Data Collection and Annotation: 1-2 months
- Model Development and Training: 2-4 months

- User Interface Development: 2-3 months
- Integration and Testing: 2-3 months
- User Education and Documentation: Ongoing

5.1.2 Project Resources

People - 4

- Hardware Resources:
  - System : Intel i3 3rd Ghz.
  - Hard Disk : 512 GB.
  - Floppy Drive : 44 Mb.
  - Monitor :15VGAColour.
  - Mouse : opticle
  - Ram : 4 GB
- Software Resources:
  - Image Processing Software
  - Machine Learning Algorithms
  - Training Data
  - Detection Algorithm
  - User Interface
  - Reporting and Alerting
  - Integration
  - Updates
  - Maintenance

5.2 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS

Analyzing risks in the context of NP-hard problems, such as car damage detection, often involves identifying computational complexity and efficiency-related concerns.

5.2.1 Risk Identification

Here are some risks associated with NP-hard analysis in car damagedetection: Analyzing risks in car damage detection with NP-hard analysis:

1. Algorithm Complexity: NP-hard algorithms may be computationally expensive.
2. Scalability: Handling larger datasets efficiently is a challenge.
3. Real-time Processing: Achieving real-time results can be difficult.
4. Resource Constraints: Limited resources in edge devices canbelimiting.
5. Parameter Tuning: Manual parameter tuning is time-consuming.
6. Data Preprocessing: Preprocessing large datasets is computationally intensive.
7. False Positives/Negatives: Complex analysis may result in errors.
8. Interoperability: Integration with existing systems can be complex.
9. Algorithm Maintenance: Maintenance and updates are crucial.
10. Data Storage/Retrieval: Efficient storage and retrieval of image dataisessential.

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

Types of Damages	Likelihood	Severity	Risk Level
Scratches	Low	Low	Low
Dents	Medium	Medium	Medium
Paint Damage	Low	Medium	Medium
Broken Windows	Low	High	High
Frame Damage	Low	High	High

Table 5.1: Risk Table

Impact Level	Description
Low	Minimal consequences; easily manageable
Medium	Moderate consequences; requires attention.
High	Significant consequences; demands immediateaction.

Table 5.2: Risk Impact

Risk Category	Risk Description	Impact Level
Technical	Detection system accuracy falls below acceptable levels	High
Operational	Data collection errors lead to false positives/negatives.	Medium
Environmental	Adverse weather conditions hinder detection accuracy.	Medium
Maintenance	Neglected system maintenance affects performance	High
Integration	Difficulty integrating the system with existing processes	Medium
Data Security	Unauthorized access to sensitive data	High
Privacy	Violation of privacy regulations	High
Calibration	Inaccurate calibration of detection equipment	Medium

Table 5.3: Performance Table

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

Risk ID	1
Risk Description	Inaccurate or insufficient training data
Category	Data
Source	Data Collection
Probability	High
Impact	High
Response	Improve dataset quality
Strategy	Conduct thorough data validation and augmentation
Risk Status	Mitigated

Risk ID	2
Risk Description	Model overfitting
Category	Algorithmic
Source	Model Development
Probability	Medium
Impact	Medium
Response	Regular model validation
Strategy	Implement regularization techniques
Risk Status	Mitigated

Risk ID	3
Risk Description	Changes in regulatory requirements
Category	External
Source	Government Regulations
Probability	Low
Impact	Medium
Response	Monitor regulatory updates
Strategy	Maintain close communication with legal experts
Risk Status	Open

Risk ID	4
Risk Description	Insufficient computing resources
Category	Resource
Source	Infrastructure
Probability	High
Impact	Medium
Response	Upgrade hardware or use cloud
Strategy	Scale resources based on workload and demand
Risk Status	Open

Risk ID	5
Risk Description	Unauthorized access or data breaches compromising sensitive information
Category	Security
Source	Cybersecurity
Probability	Low
Impact	High
Response	Strengthen security measures and access controls
Strategy	Implement encryption, authentication protocols, and regular security
Risk Status	Open

Risk ID	6
Risk Description	Technical failure leading to system downtime compromising sensitive information
Category	Technical
Source	System Failure
Probability	Medium
Impact	High
Response	Implement failover mechanisms and redundancy systems
Strategy	Develop backup systems and establish protocols for rapid response and recovery in the event of technical failures.
Risk Status	Open

Risk ID	7
Risk Description	Lack of scalability resulting in system performance degradation as the dataset grows
Category	Technical
Source	System Design
Probability	Medium
Impact	High
Response	Implement scalable architecture and infrastructure
Strategy	Design the system to accommodate increasing data volumes and user loads by leveraging scalable cloud services and distributed computing technologies.
Risk Status	Open

5.3 PROJECT SCHEDULE

5.3.1 Project task set

Major Tasks in the Project stages are:

1. Data Preprocessing
2. Model Selection
3. Model Training
4. Model Evaluation
5. Deployment
6. Inference and Reporting
7. Feedback Loop and Maintenance
8. Integration
9. Ethical Considerations
10. Testing and Quality Assurance
11. Scale and PerformanceOptimization

5.3.2 Task network

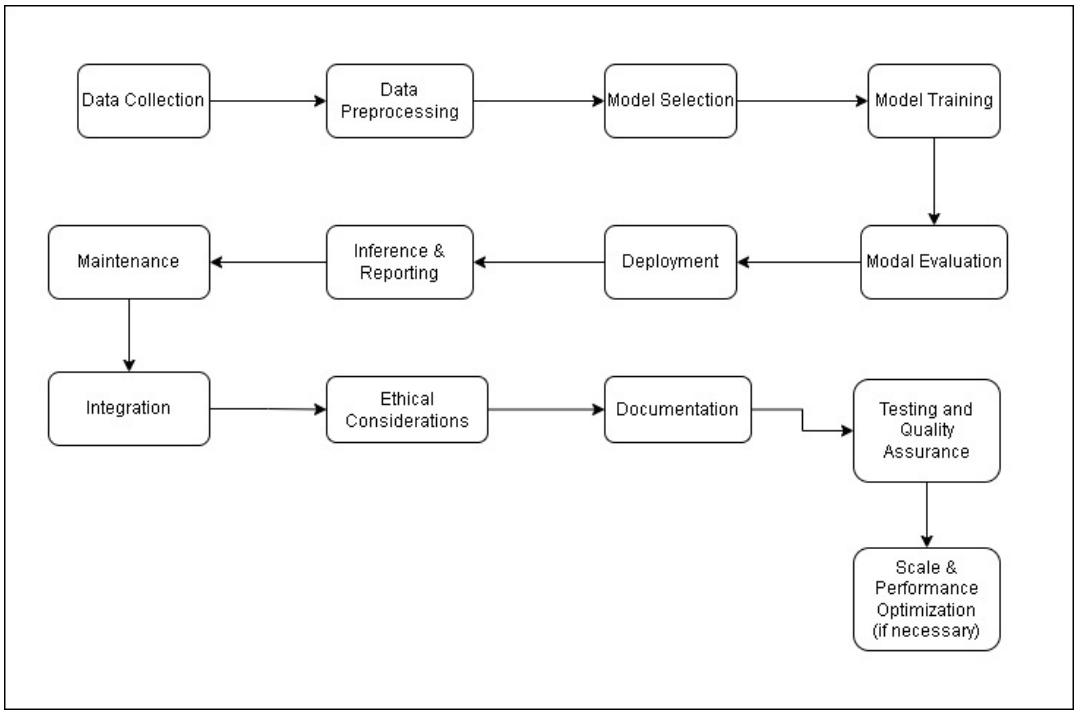


Figure 5.1: Task Network Diagram

5.4 TEAM ORGANIZATION

5.5 TEAM STRUCTURE

- **Project Coordinator:** Geeta Hade
  Responsibilities: Overseeing project management, resource allocation, and team member communication.

- **ML Developer:** Geeta Hade, Apurva Kshirsagar, Nishant Khandhar
  Responsibilities: Develops machine learning algorithms and fine-tunes models for car damage detection.

- **Frontend Developer:** Nishant Khandhar, Atharva Kasar
  Responsibilities: Designed UI for the project and integrated with ML model

- **Quality Assurance Tester:** Atharva Kasar, Apurva Kshirsagar
  Responsibilities: Designs test cases and ensures software quality through rigorous testing.

5.5.1 Management reporting and communication

Meet	Date	Project Work
1	04/08/23	First meet with guide
2	18/08/23	Discuss about various ideas
3	15/09/23	Understanding the concept of computer vision
4	29/09/23	Preparing a presentation
5	06/10/23	Study of various methods to solve a problem
6	13/10/23	Presentation on Methodologies
7	20/10/23	Presentation on Synopsis
8	31/10/23	Submit Rough copy of synopsis
9	12/01/24	Presented 1st module of the project
10	26/01/24	Presented 2nd module of the project
11	09/02/24	Completed UI for the project
12	29/02/24	Integrated module 1 and 2 with the UI
13	15/03/24	Presented 3rd module of the project
14	29/03/24	Presented 4th module of the project
15	12/04/24	Integrated module 3 and 4 with the UI
16	19/04/24	Debugged and tested the entire project
17	02/05/24	Presentation of Report and submit rough copy of report

Table 5.4: Progress Report

CHAPTER 6

SOFTWARE REQUIREMENT SPECIFICATION

6.1 INTRODUCTION

6.1.1 Purpose and Scope of Document

The Software Requirements Specification (SRS) document serves the primary purpose of providing a precise and comprehensive definition of the requirements for the car damage detection software. It is a critical reference for all stakeholders involved in the project, offering a clear and detailed roadmap for the software's development, testing, and deployment.The SRS document covers a wide spectrum of aspects. It includes functional requirements, outlining the expected behavior of the software, and non-functional requirements, specifying constraints related to performance, security, and usability. Additionally, the document details system architecture, use cases, data requirements, testing and quality assurance procedures, and the acceptance criteria for the successful deployment of the software. This comprehensive scope ensures that all essential elements are accounted for, aligning the project's objectives with its eventual execution.

6.1.2 Overview of responsibilities of Developer

- Writing and maintaining code for the car damage detection software.
- Collaborating with the project team to define software requirements.
- Designing software components and system architecture.
- Implementing and testing software features.
- and troubleshooting code issues.
- Ensuring code quality and adherence to coding standards.
- Integrating external libraries or APIs as needed.
- Conducting code reviews and providing feedback to team members.
- Staying updated with industry best practices and technologies.
- Documenting code and system architecture for reference.

- Assisting in the deployment and maintenance of the software.
- Resolving software-related issues and supporting users when needed.
- Adhering to project timelines and milestones.

6.2 USAGE SCENARIO

The Car Damage Detection system finds application in various scenarios, offering a versatile solution for multiple stakeholders. Insurance companies can expedite claims processing by allowing policyholders to upload images of damaged vehicles for quick and precise assessments. Prospective car buyers benefit from pre-purchase inspections, ensuring transparency in used car transactions. Fleet managers can maintain their vehicles with ease by routinely assessing their condition through image analysis. Rental car agencies streamline the inspection process, while vehicle sellers enhance their listings with damage assessment reports. These scenarios showcase the system's adaptability, improving efficiency and reliability in the automotive industry.

6.2.1 User profiles

1. User (Uploader)
    - Action: Provides input data (image) to the car damage detection system.
    - Role: Initiates the process by uploading an image of a damaged car.
    - Relationship: Initiates the ”Process Image Data” use case.
2. System
    - Role: Processes uploaded image data, identifies damages, generates reports, and displays results.
    - Action: Process Image Data
3. Identify Damage
    - Role: Analyzes the processed image to identify types and locations of damages.
4. Generate Report
    - Role: Compiles a detailed report based on the identified damages.
5. Display Report
    - Role: Presents the generated report to the user for review and action.
6. Relationships
    - Receives input from: User (Uploader)
    - Provides output to: User (Uploader)

6.2.2 Use-cases

1. User (Uploader)
    - Action: Upload Image
    - Role: Provides input data (image) to the car damage detection system.
    - Interactions: Initiates the process by uploading the image to the system.
2. System
    - Action: Process Image Data
    - Role: Preprocesses the uploaded image data for analysis
3. Identify Damage
    - Role: Analyzes the preprocessed image data to detect and categorize damages.
4. Generate Report
    - Role: Compiles a detailed report based on the identified damages.
5. Display Report
    - Role: Presents the generated report to the user for review andaction.

Sr No.	Use Case	Description	Actors
1	Upload Image	User uploads a damaged car image.	User
2	Process Image	Preprocesses uploaded image data.	System
3	Identify Damage	Identifies types and locations of damages.	System
4	Generate Report	Generates a detailed damage report.	System
5	Display Report	Displays the generated report.	User

Table 6.1: Use Cases

6.2.3 Use Case View

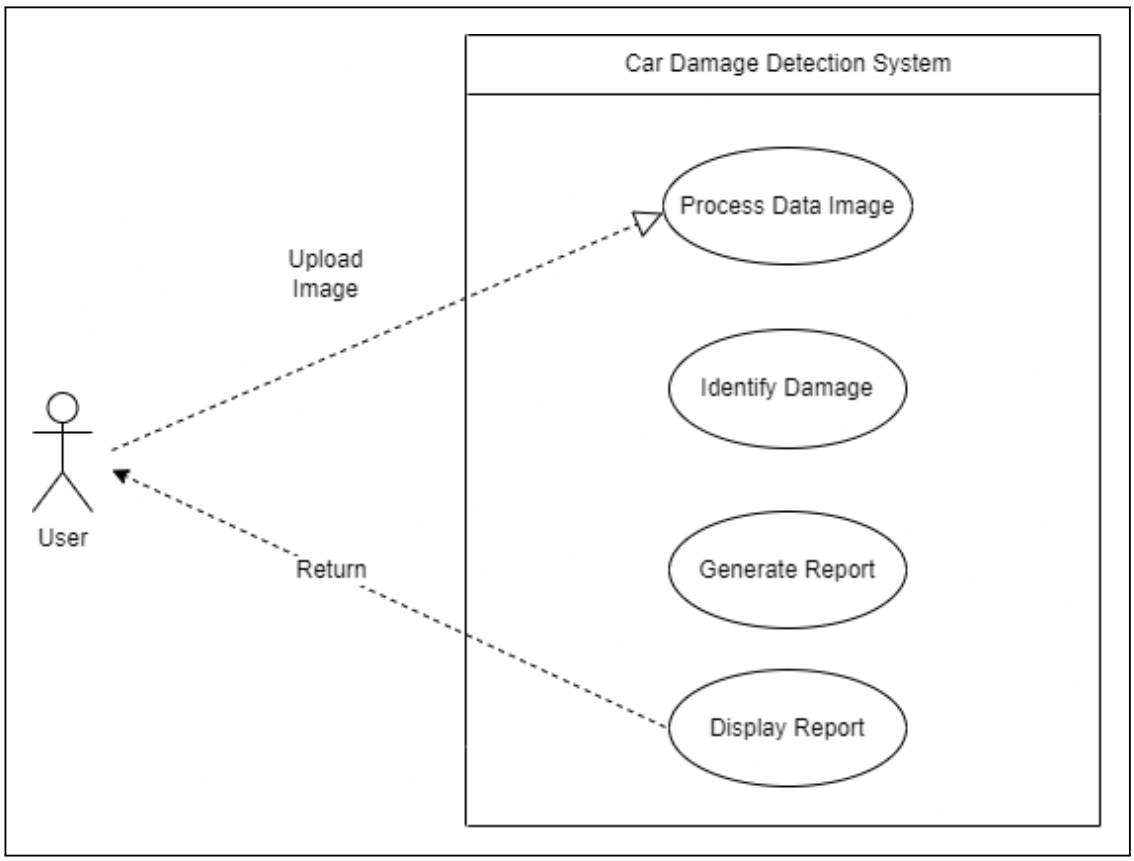


Figure 6.1: Use-Case Diagram

6.3 DATA DESCRIPTION

6.3.1 User profiles

1. Upload Image

    - Input data: Uploaded image(s) of damaged cars.
    - Description: Image data captured from various angles showing the exterior of vehi- cles with visible damages such as scratches, dents, or other types of car body damage
    - Format: JPEG, PNG, or other common image formats

2. Process Image Data

    - Input Data: Uploaded image(s) of damaged cars
    - Intermediate Data: Preprocessed image data
    - Description: Raw image data undergoes preprocessing steps to enhance quality, nor- malize colors, and prepare for damage detection algorithms.
    - Format: Processed image data in memory or storage

3. Identify Damage

    - Input Data: Analyzes the processed image to identify types and locations of damages.
    - Output Data: Detected damage types and locations.
    - Description: Image analysis algorithms analyze preprocessed data to identify specific types (e.g., scratches, dents) and locations of damages on vehicles.
    - Format: Structured data (e.g., coordinates, classifications) representing identi- fied damages.

4. Generate Report

    - Input Data: Detected damage types and locations
    - Output Data: Detailed report containing damage information.
    - Description: Based on the identified damages, a report is generated summarizing the types, locations, and severity of damages detected
    - Format: Structured report format (e.g., PDF, HTML) containing textual descrip- tions, images, and annotations.

5. Display Report:

    - Input Data: Generated report data.
    - Output Data: Displayed report for user review.
    - Description: The generated report is displayed to the user for review, enabling them to assess the detected damages and take appropriate actions (e.g., insur- ance claims, repair planning).
    - Format: Visual display on user interface (web application, mobile app) with in- teractive elements for navigation and review

6.3.2 Data objects and Relationships

Data Objects (Entities) and Attributes:

1. Image:

    - ImageID (Primary Key)
    - ImageData (Binary or reference to image file)
    - UploadTimestamp
    - UploaderID (Foreign Key referencing User)

2. Damage:

    - ImageID DamageID (Primary Key)
    - ImageID (Foreign Key referencing Image)
    - Type (e.g., Scratch, Dent, Crack)
    - Location (Coordinates or description)
    - Severity (Low, Medium, High)

3. User:

    - UserID (Primary Key)
    - Username
    - Email
    - Role (Uploader, Technician, Manager, etc.)

4. Report:

    - ReportID (Primary Key)
    - ImageData (Binary or reference to image file)
    - ReportDate
    - GeneratedBy (UserID of the system user generating the report)
    - Details (Textual description of damages and recommendations)

Relationships:

1. **Image - User (Uploader) Relationship:**

    - One-to-Many relationship (One User can upload Multiple Images)
    - Foreign Key: UploaderID in Image referencing UserID in User

2. **Image - Damage Relationship:**

    - One-to-Many relationship (One Image can have Multiple Damages)
    - Foreign Key: ImageID in Damage referencing ImageID in Image

3. **Report - Image Relationship:**

   - One-to-One relationship (Each Report is generated for One Image)
   - Foreign Key: ImageID in Report referencing ImageID in Image

6.4 FUNCTIONAL MODEL AND DESCRIPTION

Functional Model for Car Damage Detection System:

1. **Upload Image Function:**
    - **Description:** Allows users to upload images of damaged cars.
    - **Dependencies:**
        - **Class Hierarchy:**
            * ImageUploader (Base class)
            * User (Subclass)
        - **Data Flow:**
            * User interacts with UI to select and upload image files.
            * Uploaded image data is stored and processed for damage detection.
2. **Process Image Data Function:**
    - **Description:** Preprocesses uploaded image data for damage analysis.
    - **Dependencies:**
        - **Class Hierarchy:**
            * ImageProcessor (Base class)
            * DamageAnalyzer (Subclass)
        - **Data Flow:**
            * Received image data is normalized and enhanced (e.g., resizing, color adjustment).
            * Preprocessed data is passed to damage analysis algorithms.
3. **Identify Damage Function:**
    - **Description:** Analyzes preprocessed image data to identify types and locations of damages.
    - **Dependencies:**

- **Class Hierarchy:**
    - DamageDetector (Base class)
    - DamageClassifier (Subclass)
- **Data Flow:**
    - Image features are extracted and analyzed using machine learning models.
    - Detected damages (e.g., scratches, dents) are categorized and localized.

4. **Generate Report Function:**

- **Description:** Compiles a detailed report based on identified damages.
- **Dependencies:**
    - **Class Hierarchy:**
        - ReportGenerator (Base class)
        - DamageReport (Subclass)
    - **Data Flow:**
        - Damage information (types, locations) is structured into a report format.
        - Report includes textual descriptions, image annotations, and severity levels.

5. **Display Report Function:**

- **Description:** Presents the generated report to users for review and action.
- **Dependencies:**
    - **Class Hierarchy:**
        - ReportViewer (Base class)
        - UserInterface (Subclass)
    - **Data Flow:**
        - Generated report is displayed on the user interface (web, mobile app).

- Users can interact with the report to view detailed damage information and make decisions (e.g., insurance claims, repair plans).

Class Hierarchy and Dependencies:

1. **Base Classes:**
   - **ImageUploader:** Manages image upload functionality.
   - **ImageProcessor:** Handles image preprocessing tasks.
   - **DamageDetector:** Implements damage detection algorithms.
   - **ReportGenerator:** Constructs detailed reports based on detected damages.
   - **ReportViewer:** Displays generated reports on the user interface.
2. **Subclasses:**
   - **User:** Inherits from ImageUploader for user-specific interactions.
   - **DamageAnalyzer:** Extends ImageProcessor for damage analysis functionalities.
   - **DamageClassifier:** Specializes DamageDetector for damage classification tasks.
   - **DamageReport:** Subclass of ReportGenerator tailored for damage reports.
   - **UserInterface:** Implements specific UI components for report display and interaction.

6.4.1 Activity Diagram:

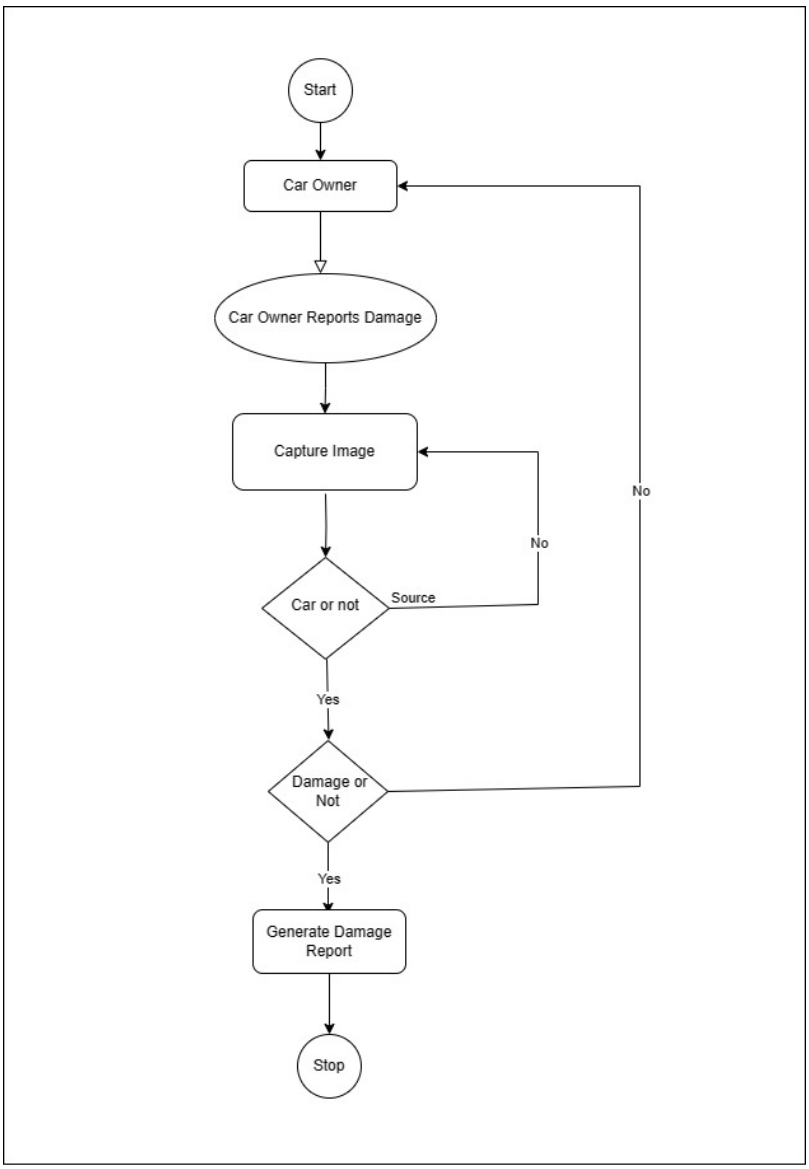


Figure 6.2: Activity Diagram

6.4.2 Non Functional Requirements:

- Interface Requirements
- Performance Requirements
- Software quality attributes such as availability [related to Reliability], modifiability [includes portability, reusability, scalability] , performance, security, testability and usability[includes self adaptability and user adaptability]

6.4.3 State Diagram:

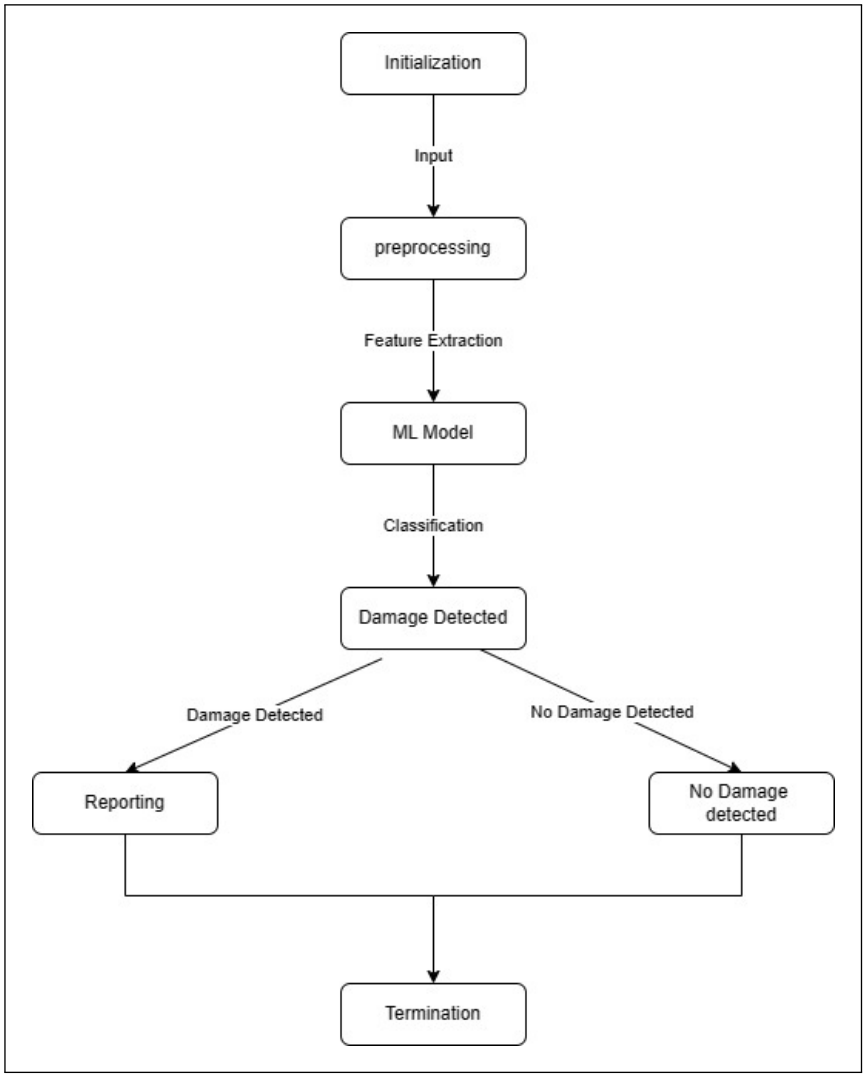


Figure 6.3: State Transition Diagram

6.4.4 Design Constraints

1. Image Quality and Variability
2. Scalability
3. Real-Time Processing
4. Accuracy and Reliability
5. Compatibility
6. User-Friendly Interface

6.4.5 Software Interface Description

1. **Home Page**

   - Description: The homepage serves as the entry point to the web app.
   - Features:
     - Logo: Displays the app logo for brand recognition.
     - Navigation Bar: Provides links to different sections of the app.
     - Upload Image Button: Allows users to upload images of damaged vehicles.
     - Footer: Contains links to important pages such as About Us, Contact, etc.

2. **Upload Image Section:**

   - Description: This page allows users to upload images of damaged vehicles for analysis.
   - Features:
     - Image Upload Form: Allows users to select and upload images.
     - Progress Bar: Indicates the upload progress.
     - Submit Button: Triggers the analysis process after image upload.

3. **Analysis Result Section:**

   - Description: Displays the results of the image analysis for car damage detection.
   - Features:
     - Image Display: Shows the uploaded image with detected damages highlighted.
     - Damage Classification: Lists the types of damages detected.
     - Damage Severity: Provides a severity rating for each detected damage.

CHAPTER 7

DETAILED DESIGN DOCUMENT USING APPENDIX A AND B

7.1 INTRODUCTION

7.2 ARCHITECTURAL DESIGN

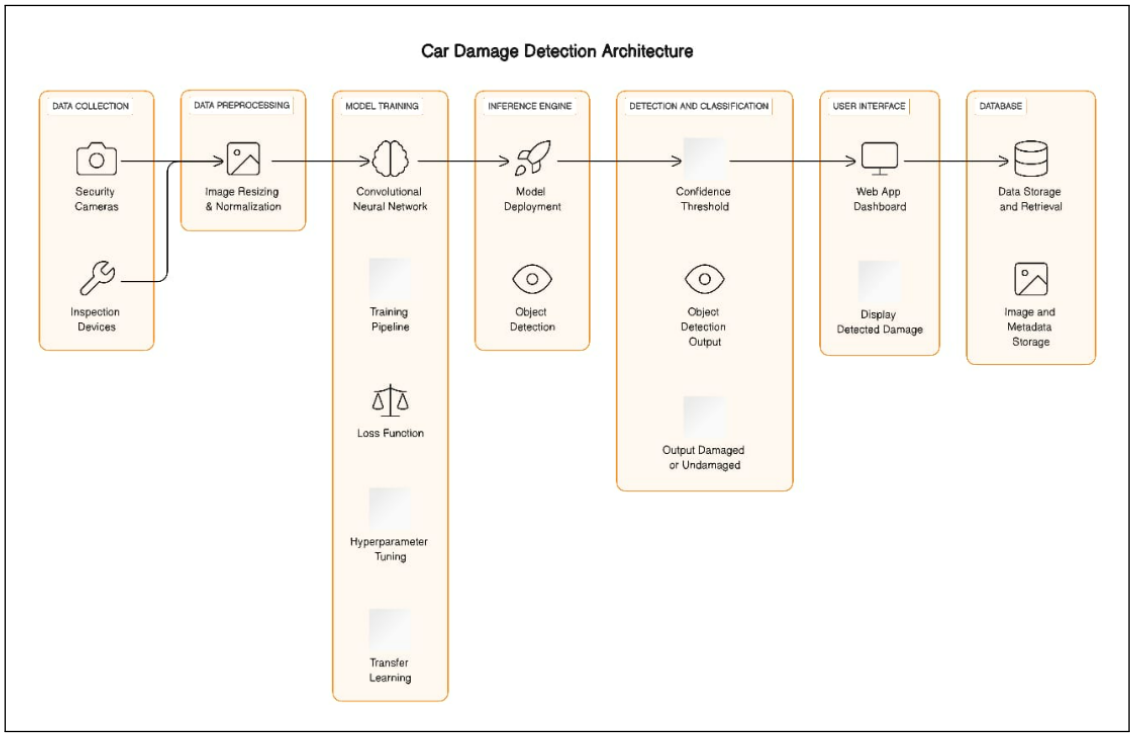


Figure 7.1: Architectural Diagram

The architectural design of the Car Damage Detection system is structured to efficiently manage its core functionalities. At a high level, the architecture can be described using subsystem design, package diagram, and deployment diagram.The system is divided into key subsystems, including the ”CarDamageDetection” subsystem responsible for core functionality. Within this subsystem, components for image processing, damage severity classification, and location tagging are encapsulated. These components work in harmony to provide a comprehensive assessment of car damage. In the package diagram, the system's functionalities are organized into logical packages. The ”Image Processing” package encompasses image handling and preprocessing components, while the ”Classification and Tagging” package includes modules for severity classification and damage location tagging. The ”User Interface” package manages interactions with users, and the ”Data Storage” package deals with database operations for storing assessment data.This architectural design optimizes the system for scalability, flexibility, and ease of maintenance.

7.3 DATA DESIGN (USING APPENDICES A AND B)

The data design encompasses a variety of data structures to support the car damage detection system. This includes internal data structures for efficient image processing, global data structures for system-wide information sharing, and temporary data structures for real-time assessment. The database design comprises tables for storing image metadata, damage assessments, and user profiles, ensuring systematic data organization. Additionally, the system utilizes standardized file formats for image storage and compatibility. This holistic approach to data design ensures streamlined data flow and efficient information management within the car damage detection system.

7.3.1 Internal software data structure

The software employs internal data structures like image containers, feature vectors, and result objects to enable effective data exchange between its components. These structures enhance communication, ensuring seamless data processing during car damage assessment.

7.3.2 Global data structure

The car damage detection system utilizes global data structures that are accessible to major components of the architecture. These structures encompass shared databases containing image metadata, assessment records, and user profiles, providing a centralized repository for system-wide information. By enabling efficient data access and retrieval across components, these global data structures foster a cohesive and synchronized operation of the software, facilitating a unified approach to car damage assessment and data management.

7.3.3 Temporary data structure

Temporary data structures are created for short-term use during the car damage assessment process. These structures include in-memory caches for storing image data during real-time analysis and result buffers for holding intermediate assessment outcomes. These temporary structures support swift data processing and assist in the

seamless flow of information within the software, improving its overall responsiveness and efficiency.

7.3.4 Database description

The application features a well-organized database system with multiple tables. The image metadata table stores image details, including source and timestamp, while the assessment records table tracks damage severity and location classifications. User profiles, such as usernames and access credentials, are stored in a separate table. This structured database design enhances data management, supporting the efficient operation of the car damage detection system.

7.4 COMPOENT DESIGN

The component diagram for the Car Damage Detection system outlines the core components and their interactions. At its center, the ”CarDamageDetectionSystem” orchestrates the following components: ”ImageProcessor” for image processing, ”SeverityClassifier” for damage severity assessment, ”LocationTagger” for damage location tagging, ”DataStorage” for database and filesystem management, and ”UserInterface” for the graphical interface and user interactions. These components work in concert to enable image processing, severity classification, location tagging, and data storage while providing a user-friendly interface for the system’s operation.

7.4.1 Class Diagram

The class diagram offers a structural overview of the Car Damage Detection system, outlining crucial classes, interfaces, and their relationships. The ”CarDamageDetection” subsystem serves as the central element, encapsulating core functionalities. The ”Image” class defines the image structure, featuring a binary data attribute for storing image content. Enumerations, ”Severity” and ”Location,” categorize damage severity levels and locations. The ”Detector” interface specifies methods for image processing, severity assessment, and location tagging. Lastly, the ”CNNModel” class represents a Convolutional Neural Network model, enabling image training, loading, and classification. This class diagram provides a concise representation of the system's components and their roles in facilitating efficient car damage assessment and classification using advanced machine learning techniques.

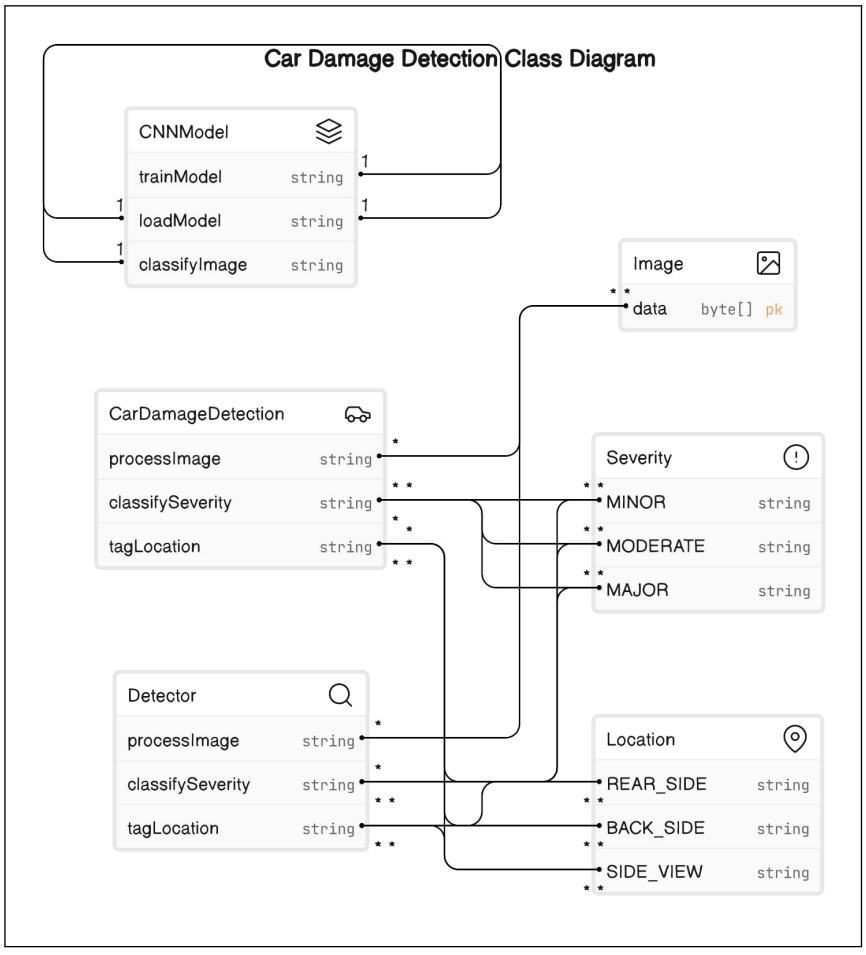


Figure 7.2: Class Diagram

CHAPTER 8

PROJECT IMPLEMENTATION

8.1 INTRODUCTION

The project involves developing a car damage detection system using computer vision techniques. The system aims to analyze images of damaged vehicles, identify types and locations of damages, and generate detailed reports for insurance claims and repairassessments.

8.2 TOOLS AND TECHNOLOGIES USED

1. TensorFlow and Keras
2. VGG16 Pre-trained Mode
3. NumPy and pandas
4. JSON
5. Jupyter Notebook
6. Atom and Pycharm
7. Django

8.3 METHODOLOGIES/ALGORITHM DETAILS

1. Data Collection and Preprocessing:
    - Gathered a dataset of damaged vehicle images with labeled damage types (scratches, dents, etc.).
    - Preprocessed images by resizing, normalizing, and augmenting to enhance model performance.
2. Transfer Learning with VGG16:
    - Implemented transfer learning using the VGG16 convolutional neural network architecture.
    - Retrained the pre-trained VGG16 model on the car damage dataset to fine-tune for damage classification

3. Logistic Regression for Binary Classification:

    - Implemented logistic regression as a baseline model for binary classification tasks (e.g., scratch vs. no scratch).

4. Model Training and Evaluation:

    - Split the dataset into training and validation sets.
    - Trained machine learning models (VGG16-based, logistic regression) on the training data.
    - Evaluated model performance using metrics such as accuracy, precision, recall, and F1-score.

5. Integration with Computer Vision Techniques:

    - Used computer vision algorithms (edge detection, contour analysis) to enhance damage localizationandboundary.

8.3.1 Pseudocode 1

```
Module 1: car or not:-
# Setup and Dependencies
import numpy as np
import json
from keras.applications import VGG16
from keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input
from collections import Counter, defaultdict
import os

# Model Preparation
vgg_model = VGG16(weights='imagenet', include_top=True)
vgg_model.save('vgg16.h5')
```

```
def prepare_image(img_path):
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    return img_array

# Prediction and Analysis
def get_prediction(img_array):
    preds = vgg_model.predict(img_array)
    top_preds = np.argsort(preds)[0][-5:][::-1]
    return top_preds, preds[0][top_preds]

def get_car_categories():
    car_categories = ['sports_car', 'convertible', 'racer', ...]
    # List of car categories from ImageNet
    return car_categories

def is_car_image(img_path):
    img_array = prepare_image(img_path)
    top_preds, _ = get_prediction(img_array)
    car_categories = get_car_categories()
    predicted_labels = [vgg_model.decode_predictions(pred)[0][0]
    [1] for pred in top_preds]
    return any(label in car_categories for label in
    predicted_labels)
```

8.3.2 Pseudocode 2

```
Module 2: Damage or not:
# Data Collection and Pre-processing
import glob
```

```
import random

def load_dataset(directory):
    image_paths = glob.glob(directory + '/*.jpg')
    random.shuffle(image_paths)
    return image_paths

def preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    return img_array

# Feature Extraction
def extract_features(img_array):
    features = vgg_model.predict(img_array)
    return features.flatten()

# Model Training and Evaluation
def train_logistic_regression(features, labels):
    from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score, confusion_matrix

    X_train, X_test, y_train, y_test = train_test_split(features,
    labels, test_size=0.2, random_state=42)
    classifier = LogisticRegression()
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
```

```
    conf_matrix = confusion_matrix(y_test, y_pred)
    return accuracy, conf_matrix

# Model Deployment and Inference
def predict_damage_status(img_path):
    img_array = preprocess_image(img_path)
    features = extract_features(img_array)
    is_damaged = logistic_classifier.predict([features])[0]
    return is_damaged
```

8.3.3 Pseudocode 3

```
Module 3: location:
# Setup and Dependencies
import numpy as np
import json
from keras.applications import VGG16
from keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input
from collections import Counter, defaultdict
import os

# Model Preparation
vgg_model = VGG16(weights='imagenet', include_top=True)
vgg_model.save('vgg16.h5')

def prepare_image(img_path):
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    return img_array
```

```
# Prediction and Analysis
def get_prediction(img_array):
    preds = vgg_model.predict(img_array)
    top_preds = np.argsort(preds)[0][-5:][::-1]
    return top_preds, preds[0][top_preds]

def get_location_categories():
    location_categories = ['garage', 'street', 'parking_lot', ...]
    # List of car location categories
    return location_categories

def identify_car_location(img_path):
    img_array = prepare_image(img_path)
    top_preds, _ = get_prediction(img_array)
    location_categories = get_location_categories()
    predicted_labels = [vgg_model.decode_predictions(pred)[0][0]
    [1] for pred in top_preds]
    location = [label for label in predicted_labels if label in
    location_categories]
    return location
```

8.3.4 Pseudocode 4

```
Module 4: severity:
# Data Processing
import h5py
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
```

```
# Data Loading
def load_data():
    with h5py.File('features.h5', 'r') as hf:
        features = np.array(hf.get('features'))
        labels = np.array(hf.get('labels'))
    return features, labels

# Data Splitting
features, labels = load_data()
X_train, X_test, y_train, y_test = train_test_split(features,
labels, test_size=0.2, random_state=42)

# Model Training
classifier = LogisticRegression()
classifier.fit(X_train, y_train)

# Model Evaluation
y_pred = classifier.predict(X_test)
accuracy = classifier.score(X_test, y_test)
conf_matrix = confusion_matrix(y_test, y_pred)
```

8.4 VERIFICATION AND VALIDATION FOR ACCEPTANCE

1. Verification

    - Model Implementation Verification:
        - Code Review: Conduct a thorough review of the implemented code to ensure it aligns with the proposed methodology and design.
        - Unit Testing: Perform unit tests on individual modules (e.g., image preprocessing, feature extraction, classification) to validate their correctness.

- Data Verification:
  - Data Consistency: Validate that the training and testing datasets are correctly preprocessed and formatted for use with the model.
  - Label Verification: Ensure that labels (e.g., car or not, damaged or not) are accurately assigned and consistent with the ground truth.

2. Verification

   - Model Performance Evaluation:
     - Accuracy Assessment: Evaluate the overall accuracy of the model in correctly identifying car images and distinguishing between damaged and undamaged cars.
     - Precision and Recall: Calculate precision (true positives / predicted positives) and recall (true positives / actual positives) to understand the model's ability to avoid false positives and false negatives.
     - Confusion Matrix Analysis: Analyze the confusion matrix to identify specific areas of model performance (e.g., sensitivity to car detection, accuracy in identifying damage).

3. Acceptance Criteria

   - Performance Metrics:
     - Define acceptance criteria based on specific performance metrics (e.g., accuracy threshold, precision, recall) that the model must achieve to be considered acceptable.
   - User Acceptance Testing (UAT):
     - Conduct user acceptance testing where stakeholders interact with the system to evaluate its usability, reliability, and effectiveness in meeting their needs.
   - Documentation and Reporting:
     - Prepare detailed documentation outlining the model's capabilities, limitations, and validation results.

CHAPTER 9

SOFTWARE TESTING

9.1 TYPE OF TESTING USED

Software testing is crucial for ensuring the quality, reliability, and performance of the car damage detection system. Here's an overview of software testing types used, test cases, and results based on the accuracy metrics provided for each module:

Testing Type	Purpose	Examples
Unit Testing	To test individual components or modules of the system in isolation.	Testing image preprocessing functions, feature extraction methods, and classifier training.
Integration Testing	To test the interactions between integrated modules.	Testing the flow of data from image preprocessing to feature extraction and classification.
System Testing	To test the entire system as a whole to ensure it meets specified requirements.	Testing the end-to-end process of uploading an image, processing it, and generating a report.
Acceptance Testing	To evaluate whether the system meets acceptance criteria and user requirements.	User acceptance testing to assess usability and effectiveness in real-world scenarios.

Table 9.1: Types of Testing

9.2 TEST CASES AND TEST RESULTS

1. Module 2 (Damaged or not):
    - Accuracy: 91%
    - Test Cases:
        - Input images of damaged and undamaged cars.
        - Validate that the model correctly predicts the presence of damage.
    - Results: The model achieved an accuracy of 91% in distinguishing between damaged and undamaged cars.
2. Module 3 (Damage Location):
    - Accuracy: 66%
    - Test Cases:
        - Input images of cars in various locations (Rear, Side, Front).
        - Validate that the model correctly predicts the location of the car.
    - Results: The model achieved an accuracy of 66% in identifying the locationofcars.
3. Module 4 (Damage Severity):
    - Accuracy: 70%
    - Test Cases:
        - Input images of cars with different levels of damage severity (e.g., minor, moderate, severe).
        - Validate that the model accurately classifies severity of car damage.
    - Results: Achieved an accuracy of 70% in correctly categorizing car damageseverity.

CHAPTER 10

RESULTS

10.1 OUTPUT AND SCREENSHOTS

10.1.1 Overall Project Accuracy:

- To calculate the overall accuracy of the project, we can use a weighted average approach based on the accuracy's of Module 2, Module 3, and Module 4:
- Weights (Assuming Equal Importance):
  Module 2: Weight = 1/3 (33.33%)
  Module 3: Weight = 1/3 (33.33%)
  Module 4: Weight = 1/3 (33.33%)
- Overall Accuracy Calculation:
  Overall Accuracy = (0.33 Accuracy of Module 2) + (0.33 Accuracy of Module 3) + (0.33 Accuracy of Module 4)
  Overall Accuracy = (0.33 91%) + (0.33 65%) + (0.33 70%)
  Accuracy = (0.33 91) + (0.33 65) + (0.33 70)
  Overall Accuracy = 30.03 + 21.45 + 23.1
  OverallAccuracy = 74.58%

10.1.2 Screenshots

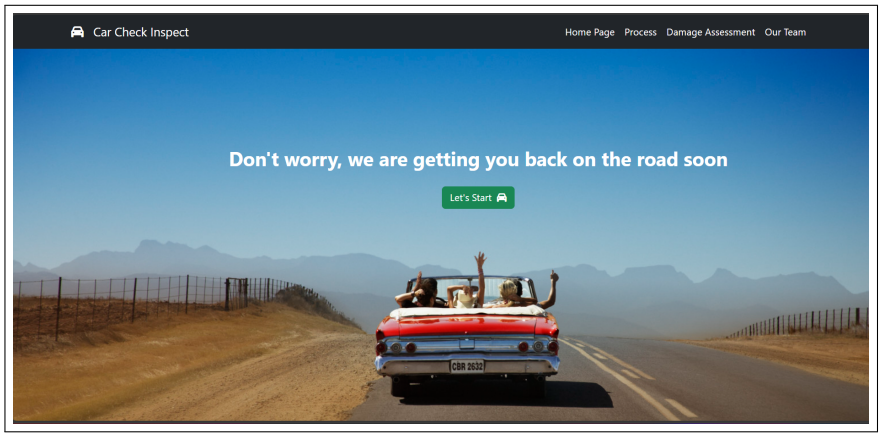


Figure 10.1: Home Page

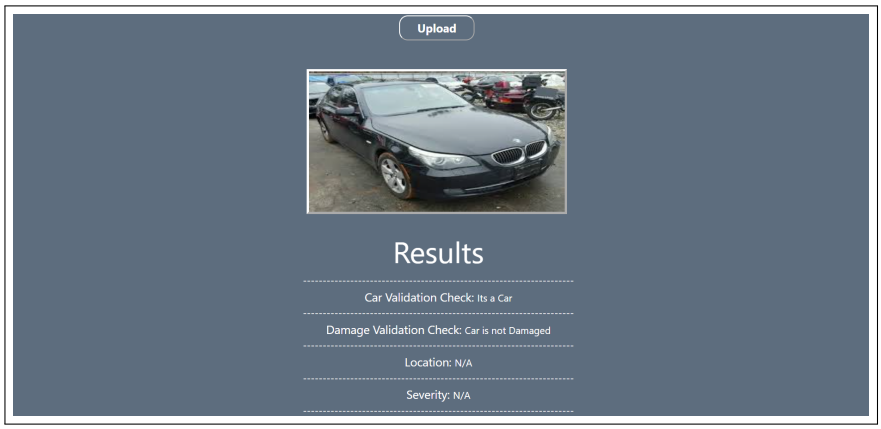


Figure 10.2: No Damage test

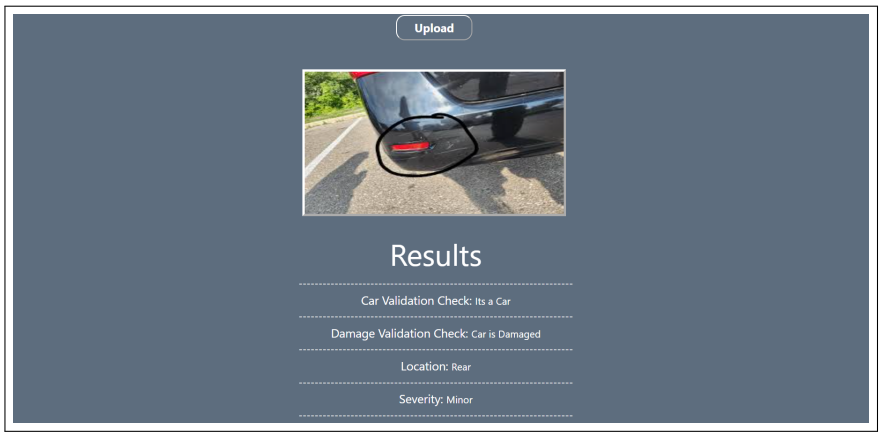


Figure 10.3: Minor Damage test

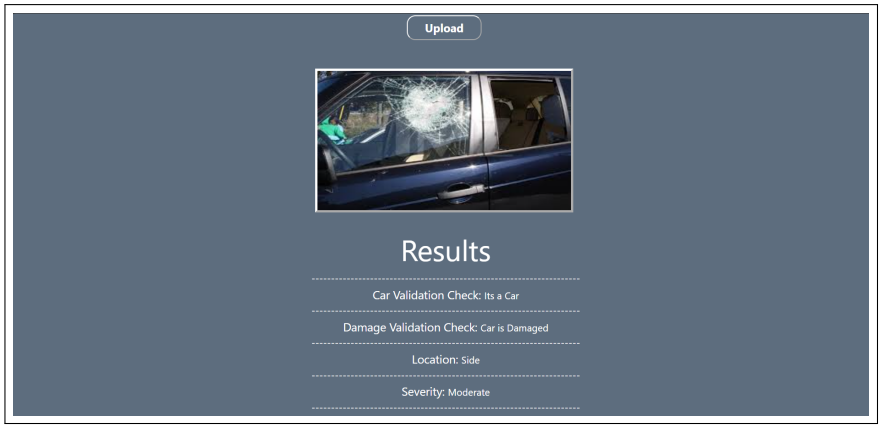


Figure 10.4: Moderate Damage test

Figure 10.5: Severe Damage test

CHAPTER 11

DEPLOYMENT AND MAINTENANCE

11.1 INSTALLATION AND UN-INSTALLATION

11.1.1 Installation

To install the software, follow these steps:

1. Download the installation package from the official website.
2. Double-click the downloaded file to initiate the installation process.
3. Follow the on-screen instructions to complete the installation.

11.1.2 Un-installation

To un-install the software, follow these steps:

1. Open the Control Panel on your computer.
2. Navigate to ”Programs” or ”Programs and Features.”
3. Find the software in the list of installed programs.
4. Click on the software and select ”Uninstall” or ”Remove.”
5. Follow the prompts to complete the un-installation process.

11.2 USER HELP

11.2.1 User Manual

A comprehensive user manual is available with detailed instructions on how to use the software’s features and functionalities. It covers installation, setup, usage guidelines, troubleshooting, and more.

11.2.2 Online Resources

Users can access additional help and support through online resources such as FAQs, forums, and documentation available on the official website.

11.2.3 Technical Support

For further assistance, users can contact our technical support team via email or phone. Our support representatives are available to address any queries or issues users may encounter during installation or usage.

CHAPTER 12

CONCLUSION AND FUTURE SCOPE

SUMMARY

The project aims to develop a robust framework for car damage detection utilizing deep learning methodologies. Leveraging convolutional neural networks (CNNs) and transfer learning, the system can accurately assess the severity and location of vehicle damage from images. Through meticulous data collection, annotation, and preprocessing, a comprehensive dataset is curated to train and validate the model. Key metrics such as accuracy, precision, recall, and F1-score are used to evaluate the system's performance. The project demonstrates the potential to streamline car damage assessment processes, enhance safety, and expedite insurance claim settlements in the automotive industry.

CONCLUSION

In conclusion, the car damage detection project presents a promising solution to address the pressing need for precise and automated evaluation techniques in the automotive sector. By harnessing the power of deep learning and computer vision, the system showcases significant advancements in identifying and classifying vehicle damage. The rigorous testing and validation processes ensure the reliability and effectiveness of the model. With further refinement and optimization, the system can be deployed in real-world scenarios to benefit car owners, insurance companies, and other stakeholders by improving efficiency and accuracy in damage assessment.

FUTURE SCOPE

Looking ahead, there are several avenues for enhancing the car damage detection system. Integration of real-time processing capabilities could enable on-the-spot damage assessment, facilitating quicker response times and decision-making in accident scenarios. Additionally, expanding the dataset to include a wider variety of car models, damage types, and environmental conditions would improve the model's robustness and generalization. Collaborations with industry partners could provide access to proprietary datasets and domain expertise, further refining the system's performance. Furthermore, exploring novel techniques such as ensemble learning.

REFERENCES

[1] M. Chen, F. Bai and Z. Gerile, ”Special Object Detection Based On Mask Rcnn,” 2021 17th International Conference on Computational Intelligence and Security (CIS), 2021, pp. 128-132, doi: 10.1109/CIS54983.20
21 .00035.

[2] M. Ye et al., ”A Lightweight Model of VGG-16 for Remote Sensing Image Classification,” in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 6916-6922, 2021, doi: 10.1109/JSTARS.2021.3090085.

[3] 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), 2021, pp. 1-5, doi: 10.1109/ICCICT50803.2021.9510069.

[4] P. M. Kyu and K. Woraratpanya, Car Damage Detection and Classification, in Proceedings of the 11th International Conference on Advances in Information Technology, Bangkok Thailand, Jul. 2020, pp. 16, doi: 10.1145/3406601
.3406651.

[5] H. Patel, hemilpatel971/Damage-car-detection, GitHub, Aug. 13, 2019.

[6] Phyu Mar Kyu, Kuntpong Woraratpanya, Car Damage Detection and Classification IAIT2020: Proceedings of the 11th International Confer- ence on Advances in Information Technology July 2020 Article No.: 46.

[7] Jeffrey de Deijn. 2018. Automatic Car Damage Recognition using Convolutional Neural Networks (2018).

[8] Sourish Dey .2019. CNN Application-Detecting Car ExteriorDamage.

[9] ”Review of deep learning: concepts, CNN architectures, challenges, applications, future directions” by Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ye Duan 2021.

[10] Overview of the development of AI dataset annotation by Bochao Ao, Bingbing Fan 2022.

[11] Vehicle Damage Severity Estimation for Insurance Operations Using In-The-Wild Mobile Images by Dimitrios Mallios ,Li Xaofie, Niall Mclaughlin ,Jesus Martinez ,Clare Galbraith 2023.

[12] Q. Zhang, X. Chang, and S. B. Bian, Vehicle-damage-detection segmentation algorithm based on improved mask RCNN, 2020.

[13] R. E. van Ruitenbeek and S. Bhulai, Convolutional neural networks for vehicle damage detection",Sep. 2022.

[14] M. Parhizkar and M. Amirfakhrian, Recognizing the damaged surface parts of cars in the real scene using a deep learning framework, Aug. 2022.

[15] H. V. Yashaswini and V. Karthik, Car damage detection and analysis using deep learning algorithm for automotive, 2019. 16)D. Widjojo, E. Setyati, and Y. Kristian, Integrated deep learning system for car damage detection and classification using deep transfer learning,2022.

ANNEXURE A

LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHMIC DESIGN

- To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix.

  Refer [**?**] for IDEA Matrix and Knowledge canvas model. Case studies are given in this book. IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

I	D	E	A
Increase	Drive	Educate	Accelerate
Improve	Deliver	Evaluate	Associate
Ignore	Decrease	Eliminate	Avoid

Table A.1: IDEA Matrix

- Project problem statement feasibility assessment using NP-Hard, NP-Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.

- input x,output y, y=f(x)

ANNEXURE B

LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN

- **Sequence Diagram**

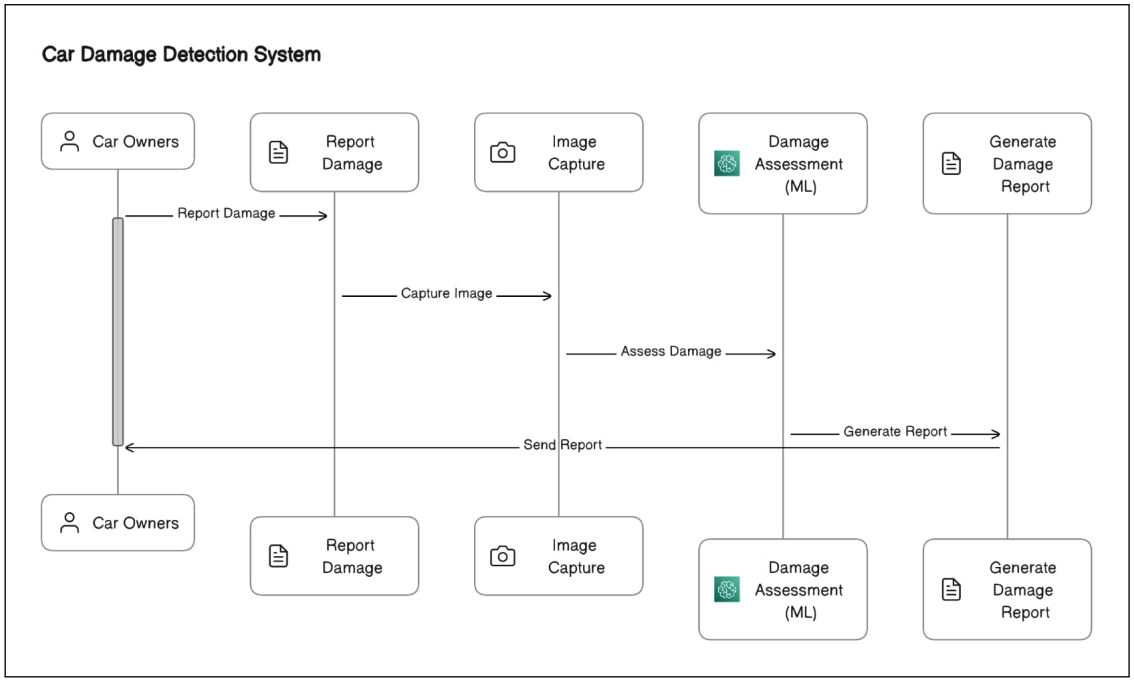


Figure B.1: Sequence Diagram

The sequence diagram for the Car Damage Detection project offers a dynamic view of how various system components and users collaborate to perform specific actions. It demonstrates the chronological order of interactions, showcasing the step-by-step communication between actors and the system. Whether it's policyholders uploading images for insurance claims, prospective buyers assessing vehicle conditions, fleet managers conducting assessments, rental car agencies managing inspections, or vehicle sellers enhancing sales listings, the sequence diagram provides a detailed depiction of the message exchanges and data flow during these processes. It serves as a valuable tool for visualizing the dynamic behavior of the system, facilitating system design, and ensuring effective communication between components and actors. The sequence diagram for the Car Damage Detection project offers a dynamic view of how various system components and users collaborate to perform specific actions. It demonstrates the chronological order of interactions, showcasing the step-by-step communication between actors and the system.

ANNEXURE C

PROJECT PLANNER

Sr. No.	Name/Title	Start Date	End Date
1	Preliminary Survey	25/08/2023	1/09/2023
2	Introduction and Problem Statement	1/09/2023	8/09/2023
3	Literature Survey	8/09/2023	15/09/2023
4	Project Statement	15/09/2023	22/09/2023
5	Software Requirement and Specification	22/09/2023	29/09/2023
6	System Design	29/09/2023	6/10/2023
7	Partial Report Submission	6/10/2023	6/10/2023
8	Architecture Design	6/10/2023	13/10/2023
9	Implementation Plan	13/10/2023	13/10/2023
10	Deployment Plan	13/10/2023	20/10/2023
11	Testing Plan	20/10/2023	27/10/2023
12	Paper Publish	9/10/2023	19/10/2023
13	Report Submission	31/10/2023	3/11/2023
14	Presented 1st and 2nd module of the project	12/01/2024	26/01/2024
15	Completed UI for the project and integrated modules	26/01/2024	29/02/2024
16	Presented 3rd and 4th module of the project	29/02/2024	29/03/2024
17	Integrated module 3rd and 4th with the UI	29/03/2024	12/04/2024
18	Debugging and Testing	12/04/2024	19/04/2024
19	Report Submission	02/05/2024	02/05/2024

Table C.1: Project Timeline

ANNEXURE D

REVIEWERS COMMENTS OF PAPER SUBMITTED

(At-least one technical paper must be submitted in Term-I on the project design in the conferences/workshops in IITs, Central Universities or UoP Conferences or equivalent International Conferences Sponsored by IEEE/ACM)

D.0.1 Sem-1 Paper:

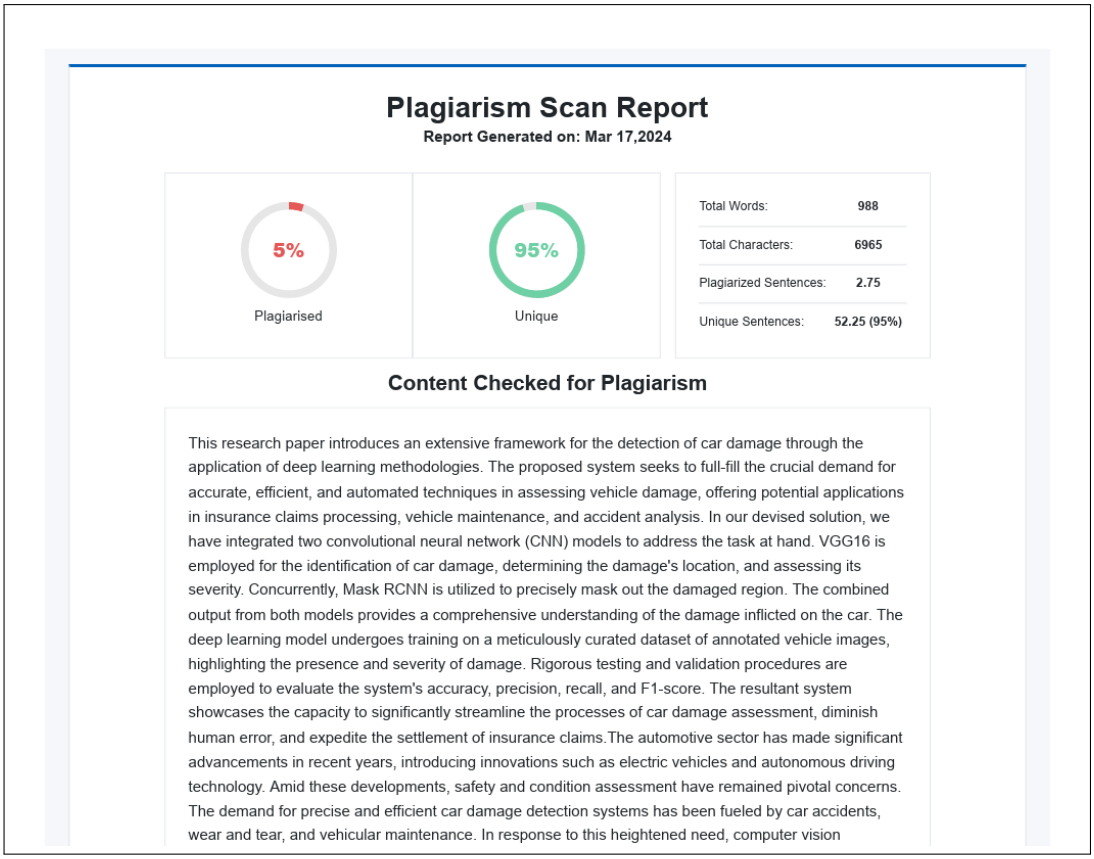
1. Paper Title: Car Damage Detection Using Computer Vision
2. Name of the Conference/Journal where paper submitted : Journal of Emerging Technologies and Innovative Research - (JETIR)
3. Paper accepted/rejected : Published
4. Review comments by reviewer :NA
5. Corrective actions if any : NA

D.0.2 Sem-2 Paper:

1. Paper Title: Car Damage Detection Using Computer Vision
2. Name of the Conference/Journal where paper submitted :
3. Paper accepted/rejected : In-Process
4. Review comments by reviewer : NA
5. Corrective actions if any : NA

ANNEXURE E

PLAGIARISM REPORT



Plagiarism Scan Report

Report Generated on: Mar 17,2024

5%
Plagiarised

95%
Unique

Total Words:	988
Total Characters:	6965
Plagiarized Sentences:	2.75
Unique Sentences:	52.25 (95%)

Content Checked for Plagiarism

This research paper introduces an extensive framework for the detection of car damage through the application of deep learning methodologies. The proposed system seeks to full-fill the crucial demand for accurate, efficient, and automated techniques in assessing vehicle damage, offering potential applications in insurance claims processing, vehicle maintenance, and accident analysis. In our devised solution, we have integrated two convolutional neural network (CNN) models to address the task at hand. VGG16 is employed for the identification of car damage, determining the damage's location, and assessing its severity. Concurrently, Mask RCNN is utilized to precisely mask out the damaged region. The combined output from both models provides a comprehensive understanding of the damage inflicted on the car. The deep learning model undergoes training on a meticulously curated dataset of annotated vehicle images, highlighting the presence and severity of damage. Rigorous testing and validation procedures are employed to evaluate the system's accuracy, precision, recall, and F1-score. The resultant system showcases the capacity to significantly streamline the processes of car damage assessment, diminish human error, and expedite the settlement of insurance claims.The automotive sector has made significant advancements in recent years, introducing innovations such as electric vehicles and autonomous driving technology. Amid these developments, safety and condition assessment have remained pivotal concerns. The demand for precise and efficient car damage detection systems has been fueled by car accidents, wear and tear, and vehicular maintenance. In response to this heightened need, computer vision

Figure E.1: Plagiarism Report

ANNEXURE F

TERM-II PROJECT LABORATORY ASSIGNMENTS

1. **Workstation Selection:**

    - Choose a workstation with sufficient computational resources (CPU, RAM, storage) to handle the development environment and potentially large datasets for testing.
    - Ensure the workstation has a stable internet connection for downloading dependencies and updates.

2. **Installations and Setup:**

    - Programming Environment (Atom and Pycharm)
    - Python Environment (Anaconda)
    - Web Development Tools (HTML, CSS, Bootstrap)
    - Machine Learning Frameworks (Tensorflow, keras)
    - Web Framework (Django)
    - Image Processing Libraries (OpenCV)

ANNEXURE G

INFORMATION OF PROJECT GROUP MEMBERS

1. Name : Nishant Khandhar
2. Date of Birth : 22/06/2002
3. Gender : Male
4. Permanent Address : Amalner, Maharashtra, India
5. E-Mail : nishantsk2002@gmail.com
6. Mobile/Contact No. :+91 7719926583
7. Placement Details : MS
8. Paper Published : JETIR

1. Name : Atharva Kasar
2. Date of Birth : 29/06/2002
3. Gender : Male
4. Permanent Address : Nashik, Maharashtra, India
5. E-Mail : akasar463@gmail.com
6. Mobile/Contact No. :+91 9373954183
7. Placement Details : Accenture
8. Paper Published : JETIR

1. Name : Apurva Kshirsagar
2. Date of Birth : 30/11/2002
3. Gender : Female
4. Permanent Address : Pune, Maharashtra, India
5. E-Mail : kshirsagarapurva30@gmail.com
6. Mobile/Contact No. :+91 9579661739
7. Placement Details : Virtual Internship
8. Paper Published : JETIR

1. Name : Geeta Hade
2. Date of Birth : 15/12/2002
3. Gender : Female
4. Permanent Address : Pune, Maharashtra, India
5. E-Mail : geetahade2018@gmail.com
6. Mobile/Contact No. :+91 9146712043
7. Placement Details : MS
8. Paper Published : JETIR

JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Car Damage Detection Using Computer Vision

Priti Warungse[1], Atharva Kasar[2], Apurva Kshirsagar[3], Geeta Hade[4], Nishant Khandhar[5]

[1]*(M.E Computer Engineering/ NBN Sinhgad Technical Institute Campus, India)*
[2]*(Computer Engineering/ NBN Sinhgad Technical Institute Campus, India)*
[3]*(Computer Engineering/ NBN Sinhgad Technical Institute Campus, India)*
[4]*(Computer Engineering/ NBN Sinhgad Technical Institute Campus, India)*
[5]*(Computer Engineering/ NBN Sinhgad Technical Institute Campus, India)*

Abstract: This research paper presents a comprehensive framework for car damage detection using deep learning techniques. The proposed system aims to address the critical need for Precise, effective, and automated techniques for evaluating vehicle damage, with potential applications in insurance claims processing, vehicle maintenance, and accident analysis. The project leverages a state-of the-art convolutional neural network (CNN) the architectural design, particularly in the case of ResNet for its outstanding feature extraction prowess and classification performance. The deep learning model is trained on a carefully curated dataset of vehicle images, annotated with labels indicating the presence and severity of damage. The project undergoes rigorous testing and validation to assess its accuracy, precision, recall, and F1-score. User feedback and user experience evaluations are considered for continuous improvement. The resulting system demonstrates the potential to significantly streamline car damage assessment processes, reduce human error, and expedite insurance claim settlements. The project undergoes rigorous testing and validation to assess its metrics encompassing accuracy, precision, recall, and F1-score.

Keywords: Computer Vision, Image Processing, Machine Learning (ML), Deep Learning (DP), Convolutional Neural Network (CNN), Feature Extraction, Object Detection, Region Of Interest (ROI), Anomaly Detection), Segmentation, Paging, Classification, Accuracy, Precision, Recall, F1-Score, Training and Testing Dataset, ROI Localization, Real-Time Detection, Transfer Learning, Data Augmentation

---

Date of Submission: 10-11-2023 Date of acceptance: XX-XX-2023

I. INTRODUCTION

In recent years, the car industry has made big progress with new things like electric cars and cars that can drive themselves. Amid these developments, safety and condition assessment have remained pivotal concerns. Car accidents, wear and tear, and vehicular maintenance have driven the need for precise and efficient car damage detection systems. In reaction to this increased need, computer vision technology has arisen as a ground breaking solution. This research paper delves into the domain of "Car Damage Detection using Computer Vision" to explore the state-of-the-art techniques, challenges, and real-world applications that leverage artificial intelligence and image processing to assess, classify, and locate car damages. With a focus on enhancing safety, facilitating insurance claims, supporting pre-purchase inspections, and streamlining fleet management, this paper highlights the multidimensional impact of computer vision in the automotive sector. The research discusses the underlying technologies, methodologies, and the potential for wider adoption across various automotive stakeholders, setting the stage for a comprehensive exploration of this cutting-edge field. The proposed framework is capable of predicting the type of vehicle damage i.e. either its minor damage or major damage. The proposed system is based on the machine learning algorithm as it evolving technology in artificially intelligent systems. Assessing Car Damage with Convolution Neural Networks and also utilizes the keywords ” bumper dent”, ”door dent”, etc. This paper applies Convolutional Neural Networks (CNNs) to damaged car images to assess the extent of damage - they use transfer learning to evaluate the merits of object recognition models that are available.

II. METHODOLOGY

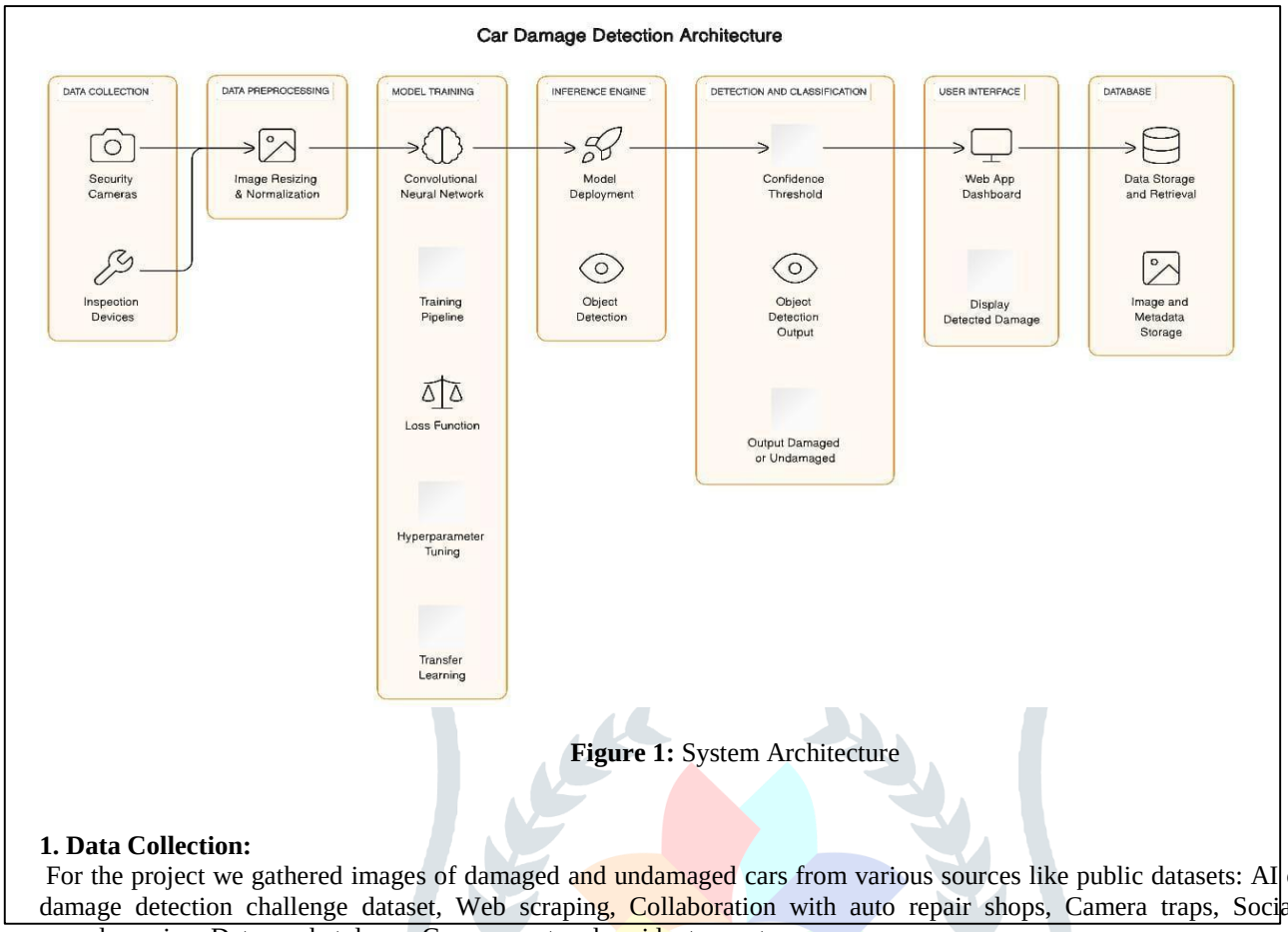


Figure 1: System Architecture

1. Data Collection:

For the project we gathered images of damaged and undamaged cars from various sources like public datasets: AI crowd AI car damage detection challenge dataset, Web scraping, Collaboration with auto repair shops, Camera traps, Social media and crowdsourcing, Data marketplaces, Government and accident reports.

2. Data annotation and Labeling:

We Developed clear and detailed labeling guidelines that define what constitutes car damage in the context of project that are specific about the types of damage you want to detect (e.g., scratches, dents, major accidents). Tools used for image annotation software are Labelbox, RectLabel, or even simple drawing software. Annotators go through each image in the dataset and mark the damaged areas by drawing bounding boxes or other annotation shapes around them. In the case of undamaged images, they may simply label them as "undamaged."

If project involves detecting different types of car damage, instructions are given to categorize the damage into specific subcategories. For example, categorize scratches, dents, and other types. We annotate the dataset by labeling each image as either "damaged" or "undamaged."

3.Data Preprocessing:

It involves preparing and cleaning the dataset to make it suitable for training a machine learning model.

Following are the steps:

a) Data Collection:

Gather a diverse dataset of car images, including damaged and undamaged examples.

b) Data Labeling:

Annotate the dataset by distinguishing between damaged and undamaged areas in the images, categorizing types of damage as needed.

c)Data Cleaning:

Check for missing or corrupted images and remove them to ensure data quality.

d)Data Splitting:

Divide the dataset into training, validation, and testing sets for model training and evaluation.

e) Data Resizing:

Resize images to a consistent dimension (e.g., 224x224 pixels) to ensure uniform input for the model.

f) Normalization:

Scale pixel values within a common range (e.g., 0 to 1) to facilitate model training.

g) Data Augmentation:

Apply techniques like rotation, flipping, and contrast adjustments to increase data diversity and improve model generalization.

h) Data Balancing:

Ensure there's a roughly equal representation of damaged and undamaged images in the training dataset to prevent class imbalance.
i)Data Encoding:
Convert categorical data (e.g., car make, model) into numerical values using methods like one-hot encoding.

4.Model Selection:
Choosing a suitable model architecture based on your dataset size and complexity is important for image classification. Convolutional Neural Networks (CNNs) are a common choice.
a. CNNs: Choose from architectures like ResNet, VGG, or Inception, and fine-tune them.
b. Object Detection: Utilize models like YOLO, Faster R-CNN, or SSD for localizing and classifying damage.
c. Transfer Learning: Consider pre-trained models for feature extraction.

5.Model Training and Testing:
We Split dataset into training and testing subsets is a crucial step in machine learning to evaluate your model's performance. A common approach is to use a train-test split, typically with a ratio of 70-80% for training and 20-30% for testing.

6.Hyperparameter Tuning:
We Experiment with hyperparameters such as learning rate, batch size, and number of epochs to find the best configuration using techniques like grid search or random search.

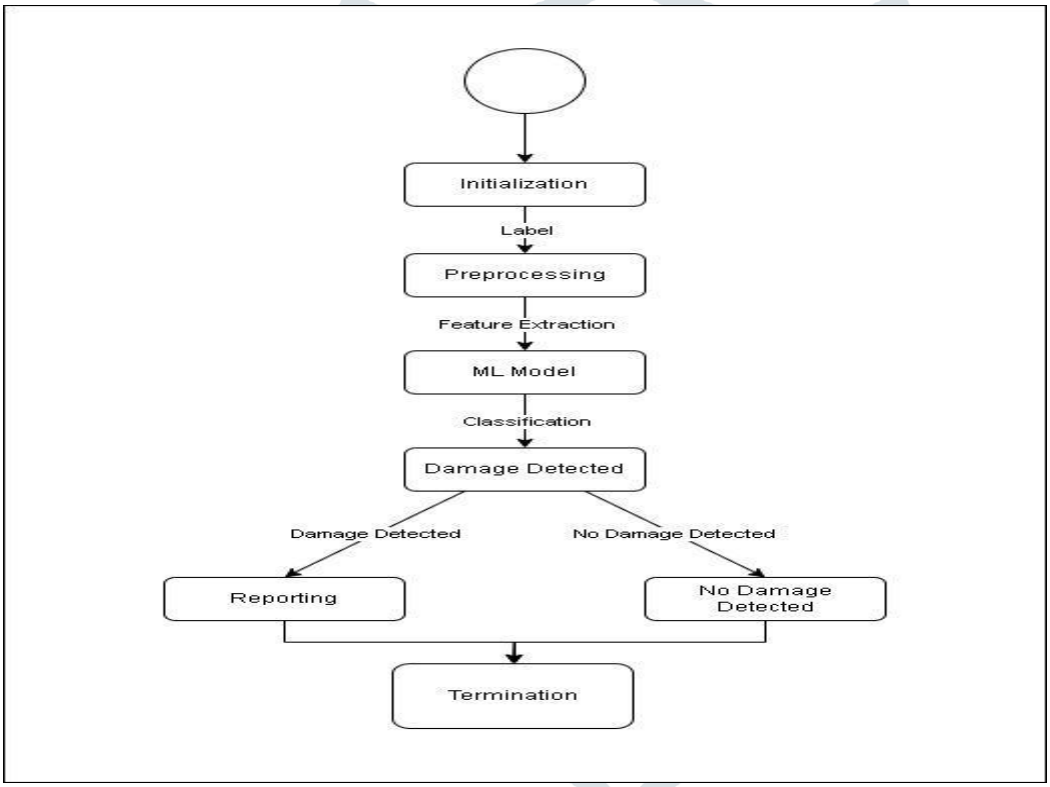


Figure 2: State Transition Diagram

III. PERFORMANCE EVALUATION

The car damage detection project aims to automate the assessment of vehicle damage, improving accuracy, efficiency, and reducing human error in the process. It benefits car owners and insurance companies by expediting claims and reducing costs. Paper present the automation process for processing insurance claims related to damaged car images using deep learning. The automatic classifier has been built using a multilayer Convolution neural network (S-CNN). The CNN architecture is used to classify the damaged images. The values in the confusion matrix are used to calculate various evaluation metrics such as accuracy, precision, recall (sensitivity), specificity, F1-score, and more, to assess the model's performance in car damage detection.
In a multi-class classification for car damage detection, the confusion matrix extends to multiple classes, representing the model's performance in distinguishing different levels of damage, if applicable.

Table 1: Confusion Matrix

Confusion Matrix	
True Positive (TP)	Model correctly predicted damaged cars.
False Positive (FP)	Model correctly predicted undamaged cars.
True Negative (TN)	Model incorrectly predicted undamaged cars as damaged (Type I error).
False Negative (FN)	Model incorrectly predicted damaged cars as undamaged (Type II error).

To improve the accuracy of the CNN model, dataset augmentation has been done. The results have been evaluated and tested on both the dataset without augmentation and with augmentation. The results obtained from the augmented dataset are much better than without the augmented dataset. The results generated by the CNN classifier would be further improved by the inclusion of transfer learning and ensemble learning and finally fine-tuned making the accuracy of the classifier up to 90%.
We monitored the change in loss function and accuracy against validation dataset by increasing the number of epochs and number of per steps inside epochs in order to verify improvement in the damage classifier. The algorithm parameters were adjusted like dropout rate, optimizer and custom layers with activation functions were introduced during the experiment. In case of vehicle damage classifier 90% accuracy is achieved. Precision and recall for "huge damage" and "no damage" outperformed those of the "medium damage" class. The f1 score of medium damage class is 87% because of its sensitivity. Medium damage contains the images with minor damage which is very close to a car that has no damage. For images with more effect, accuracy is 91% and all the
images in test data were classified correctly.

IV. CONCLUSION

The Car Damage Detection project has demonstrated the feasibility and effectiveness of using computer vision and machine learning techniques to automate car damage assessment. The system's ability to process and classify damage severity and location provides a valuable tool for both individuals and businesses in the automotive industry. The project has shown that with the right combination of image processing, machine learning models, and an intuitive "In the field of user interface development, automating tasks such as car damage assessment is not only achievable but also highly beneficial." This advancement can potentially save time, reduce human error, and streamline the entire assessment process. The successful finishing of the project paves the way for further enhancements and practical uses in the real world. It's important to continue refining the system, exploring opportunities for real-time assessment, and addressing challenges such as scale and robustness. Overall, the Car Damage Detection Project represents It significantly contributes to the computer vision field and paves the way for future advancements in automated image-based assessments.

REFERENCES

[1]. M. Chen, F. Bai and Z. Gerile, "Special Object Detection Based On Mask RCNN," 2021 17th International Conference on Computational Intelligence and Security (CIS), 2021, pp. 128-132, DOI: 10.1109/CIS54983.20 21 .00035.Ahmed, M. S., Mohammed, A. S., & Agusiobo, O. B. (2006). Development of a Single Phase Automatic Change Over Switch. *AU Journal of Technicial Report*, *10*(1), 68–74.

[2]. M. Ye et al., "A Lightweight Model of VGG-16 for Remote Sensing Image Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 6916-6922, 2021, DOI: 10.1109/JSTARS.2021.3090085.Brown, B., Guditis, J., Critical, S. D., & Competency, P. (2006). Critical-Power Automatic Transfer Systems – Design and Application. *IEEE Central TN Section*, *1*(August), 1–18.

[3]. 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), 2021, pp. 1-5, DOI: 10.1109/ICCICT50803.2021.9510069.Christian, M. (2012). Smart Phase Change-over system with AT89C52 Microcontroller. *Journal of Electrical and Electronics Engineering*, *1*(3), 31–34.

[4]. P. M. Kyu and K. Woraratpanya, "Car Damage Detection and Classification," in Proceedings of the 11th International Conference on Advances in Information Technology, Bangkok Thailand, Jul. 2020, pp. 1–6, DOI: 10.1145/3406601.3406651.

[5]. Phyu Mar Kyu, Kuntpong Woraratpanya," Car Damage Detection and Classification" IAIT2020: Proceedings of the 11th International Conference on Advances in Information Technology July 2020 Article No.: 46 Pages 1–6 https://doi.org/10.1145/3406601.3406651.

[6]. H. Patel, "hemilpatel971/Damage-car-detection," GitHub, Aug. 13, 2019.

[7]. Sourish Dey .2019. "CNN Application-Detecting Car Exterior Damage".

[8]. Jeffrey de Deijn. 2018. "Automatic Car Damage Recognition using Convolutional Neural Networks" (2018).