SAVITRIBAI PHULE PUNE UNIVERSITY

A PRELIMINARY PROJECT REPORT ON

Car Damage Detection using Computer Vision

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Atharva Kasar	B5458
Apurva Khirsagar	B5469
Geeta Hade	B5444
Nishant Khandhar	B5462

Under The Guidance of
Prof. P.B. Warungse

DEPARTMENT OF COMPUTER ENGINEERING
Sinhghad Technical Education Society's
NBN Sinhgad Technical Institutes Campus
Ambegaon (BK) Pune 411041
Savitribai Phule Pune University

Academic Year: 2023-24

DEPARTMENT OF COMPUTER ENGINEERING
Sinhghad Technical Education Society's
NBN Sinhgad Technical Institutes Campus
Ambegaon (BK) Pune 411041
Savitribai Phule Pune University
Academic Year: 2023-24

CERTIFICATE

This is to certify that the Project Entitled

Car Damage Detection using Computer Vision

Submitted by

Atharva Kasar	B5458
Apurva Khirsagar	B5469
Geeta Hade	B5444
Nishant Khandhar	B5462

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. P.B Warungse
Internal Guide
Dept. of Computer Engg.

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

Abstract

This research paper presents a comprehensive framework for car damage detection using deep learning techniques. The proposed system aims to ad- dress the critical need for accurate, efficient, and automated methods for as- sessing vehicle damage, with potential applications in insurance claims pro- cessing, vehicle maintenance, and accident analysis.The project leverages a state-of-the-art convolutional neural network (CNN) architecture, specifi- cally ResNet, for its superior feature extraction capabilities and classifica- tion 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.

Acknowledgments

Please Write here Acknowledgment.Example given as
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 ***Prof. S.P. Benadale****, Head of Computer Engineering Department, NBN Sinhgad Technical Institute 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

- Develop a robust machine learning model for accurate detection of vehicle damages. Gather and annotate a diverse dataset of vehicle images to train and validate the model. Integrate advanced computer vision techniques for efficient image analysis. Optimize the model to enable real-time processing and immediate damage assessment. Enhance model accuracy through continuous fine-tuning and feedback loops. Integrate the system seamlessly with existing processes and workflows. Design an intuitive user interface for easy interaction and result interpretation

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:
- 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
- Conferences/workshops in IITs
- Central Universities or SPPU Conferences
- IEEE/ACM Conference/Journal 2

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

1. Project Initiation
2. Data Collection and Preparation
3. Model Development
4. Software Development
5. Testing, Evaluation, and Deployment

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 promptly repaired.

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] 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.

[2] Assessing Car Damage with Convolution Neural Networks and also utilizes the keywords ”bumper dent”, ”door dent”, ”glass shatter” etc

[3] The implementation of this project is done two parts, first one consists of VGG models [10] wherein 3 models are trained using transfer learning. In the second part of implementation, Mask R-CNN [11] has been used for localizing the damage of the car.

[4] Proposed system allows the user to upload a vehicle image through the developed mobile application. We have adapted a pre-trained deep learning model to recognise and classify multiple types of vehicle damages.

[5] 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.

[6] Applies to real world datasets deep learning-based algorithms, VGG16 and VGG19, to car damage detection and evaluation. They investigate the impact of domain-specific pre-trained CNN models trained on the ImageNet dataset and fine-tuned in this paper. The researchers then use transfer learning to improve the accuracy of pre- trained VGG models, as well as other techniques. They use a blend of transfer learning and L2 regularisation to attain accurateness in damaged detection, damage localization, and damage severity. Their study indi- cates that VGG19 is better than VGG16 with an accuracy of 95.22 percent. Reference

[7] uses convolutional neural networks for identification whether a car image actually shows dam- age. They demonstrated that their transfer learning VGG16 approach has the potential to deliver excellent results when the dataset is larger and of higher quality. They correctly classified the type, location, and size of the damage with 75.1 percent,68.7 percent, and 54.2 percent accuracy, respectively.

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 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 development and testing).

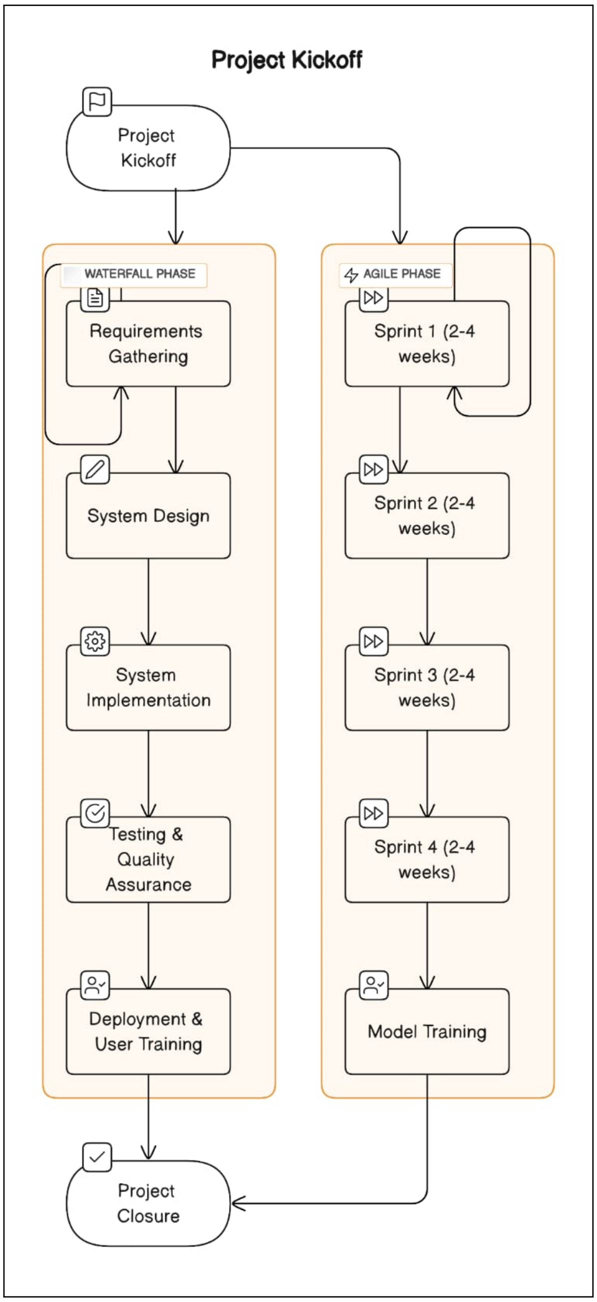

Figure 5.1: Hybrid Project Management Model

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 : 15 VGA Colour.
  - 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 damage detection: 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 can be limiting.

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 data is essential.

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 immediate action.

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

5.3 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 Performance Optimization

5.3.2 Task network

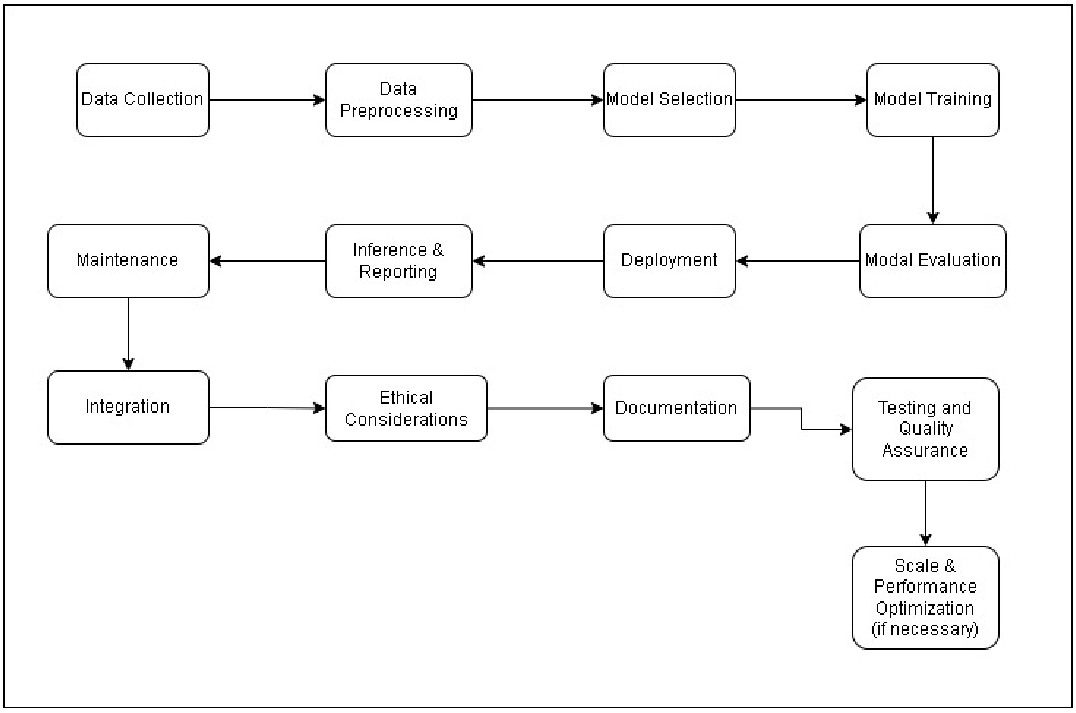

Figure 5.2: Task Network Diagram

5.4 TEAM ORGANIZATION

5.4.1 Team structure

- Atharva Kasar - Developer, Analyst

- Apurva Kshirsagar - Developer, Analyst
- Geeta Hade - Developer, Analyst
- Nishant Khandhar - Developer, Analyst

5.4.2 Management reporting and communication

Meet	Date	Project Work
1	04/08/23	First meet with guide
2	15/09/23	Discuss about various ideas
3	06/10/23	Understanding the concept of computer vision
4	31/10/23	Preparing a presentation
5	31/10/23	Study of various methods to solve a problem
6	31/10/23	Presentation on Methodologies
7	31/10/23	Presentation on Synopsis
8	31/10/23	Submit Rough copy of synopsis

Table 5.4: Progress Report

CHAPTER 6

SOFTWARE REQUIREMENT SPECIFICATION (SRS IS TO BE PREPARED USING RELEVANT MATHEMATICS DERIVED AND SOFTWARE ENGG. INDICATORS IN ANNEX A AND B)

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.

6.3 UML DIAGRAMS AND DESCRIPTION

6.3.1 Use-case diagram

The use case diagram provides a comprehensive view of the interactions between actors and the system in the Car Damage Detection application. It depicts various use cases that represent the specific functionalities and tasks performed by different user categories. Policyholders can initiate the ”File Insurance Claim” use case, allowing them to upload images for damage assessment. Prospective buyers engage in ”Perform Pre-Purchase Inspection” to evaluate the condition of vehicles they're interested in. Fleet managers use the ”Conduct Fleet Assessment” use case for routine damage checks and maintenance assessments. Rental car agencies utilize the ”Manage Rental Vehicle Inspections” use case to streamline post-rental inspections. Lastly, vehicle sellers participate in ”Enhance Sales Listings” to include damage

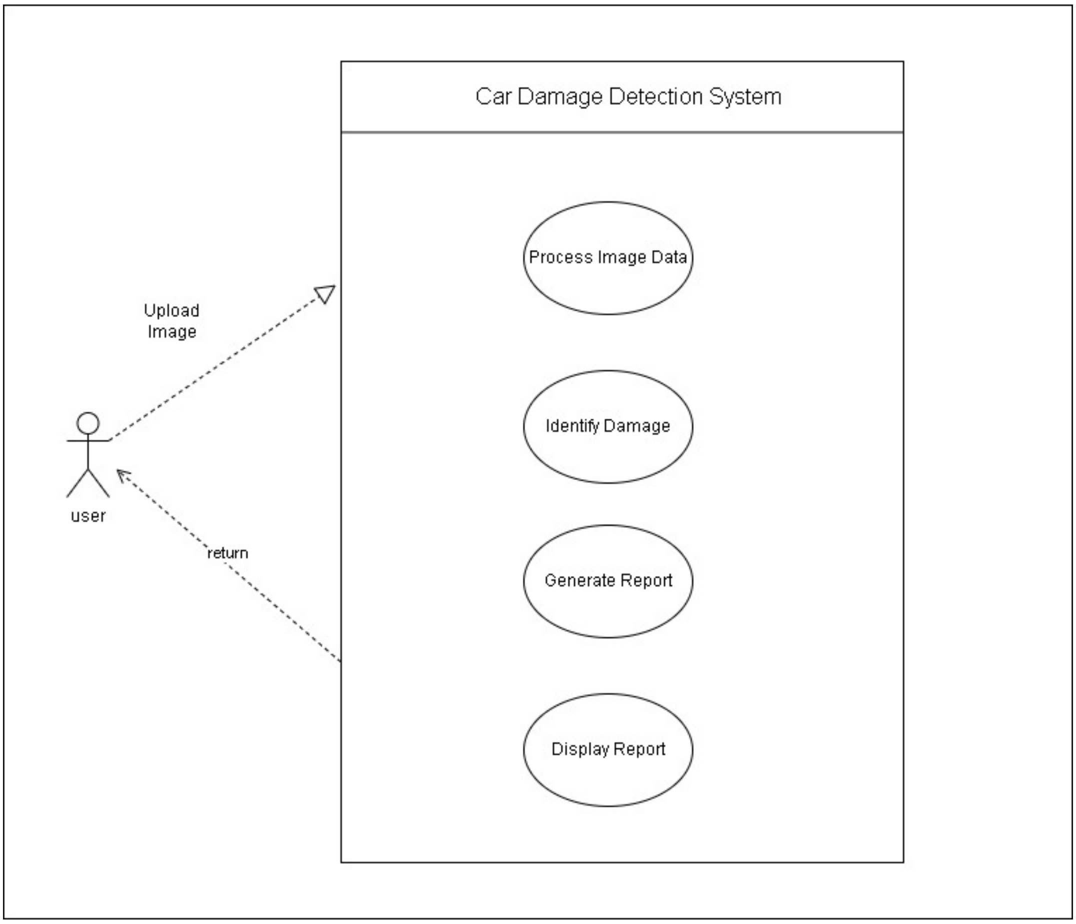

Figure 6.1: Use-Case Diagram

assessment reports in their listings. This use case diagram illustrates the system's adaptability to cater to the diverse needs of actors in the automotive industry, enhancing efficiency and reliability.

6.3.2 Class Diagram

The class diagram represents the fundamental structure of the Car Damage Detection system, illustrating key classes, interfaces, and their relationships. The ”CarDamageDetection” subsystem acts as the central component, encapsulating the essential functionalities. ”Image” is a class representing the structure of images, with a binary data attribute for storing image data. Two enums, ”Severity” and ”Location,” define various levels of damage severity and damage location, respectively. The ”Detector” interface outlines methods required for image processing, severity classification, and location tagging. Lastly, the ”CNNModel” class represents a Convolutional Neural Network model, which can be trained, loaded, and utilized for image classification.

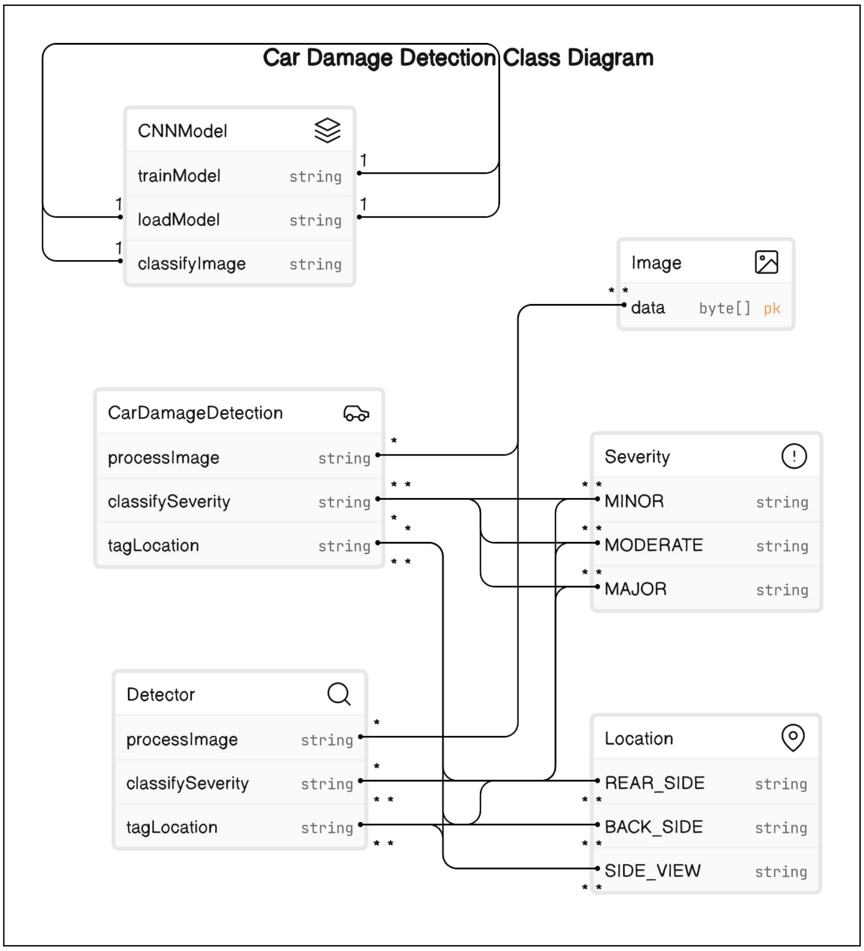

Figure 6.2: Class Diagram

This class diagram provides a comprehensive view of the system's components and their interactions, facilitating efficient car damage assessment, classification, and location tagging using a CNN model.

6.3.3 Flowchart Diagram

The flowchart diagram for the Car Damage Detection project is a visual representation of the system's dynamic behavior, illustrating the sequence of activities and interactions among users and the system. It outlines the key processes, from policyholders uploading images for insurance claims to prospective buyers inspecting vehicles, fleet managers assessing their vehicles, rental car agencies managing inspections, and vehicle sellers enhancing sales listings. This diagram captures the workflow of each user category, depicting how they interact with the system to assess car damage severity and location. It provides a clear and structured overview of the project's functionality, enhancing understanding and aiding in the system's design and implementation.

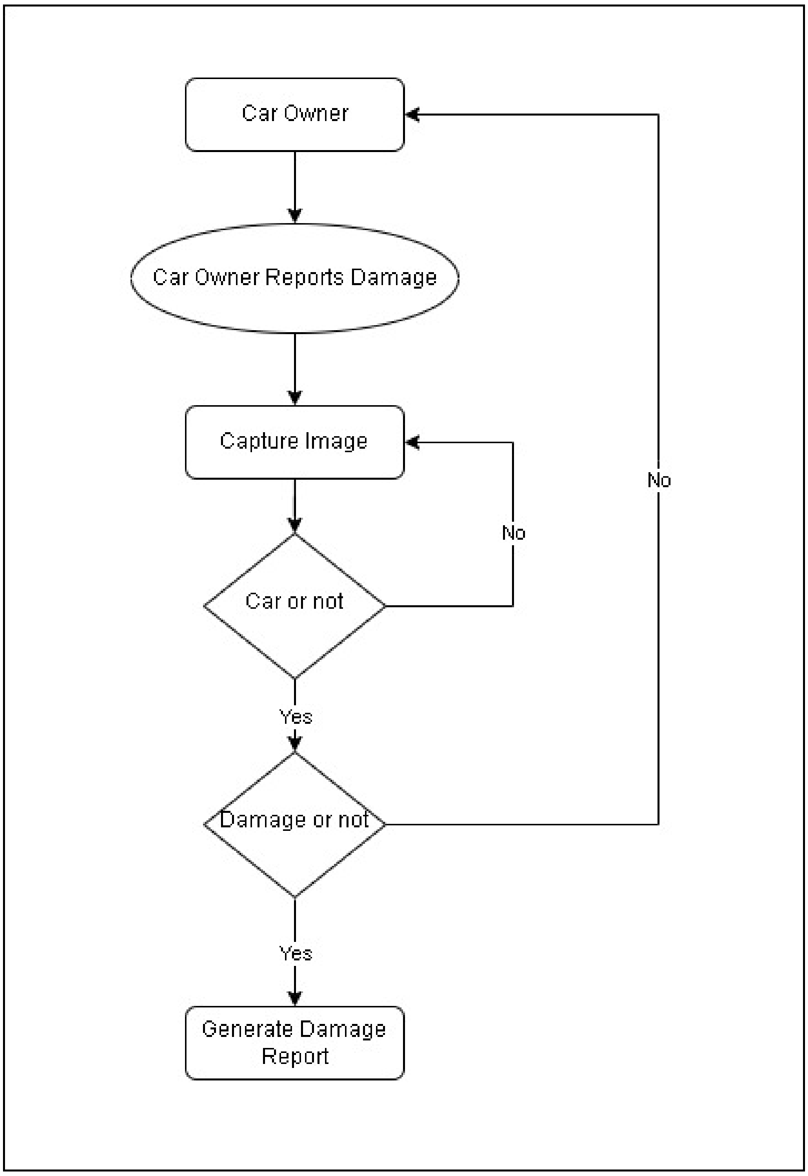

Figure 6.3: Flowchart Diagram

6.3.4 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,

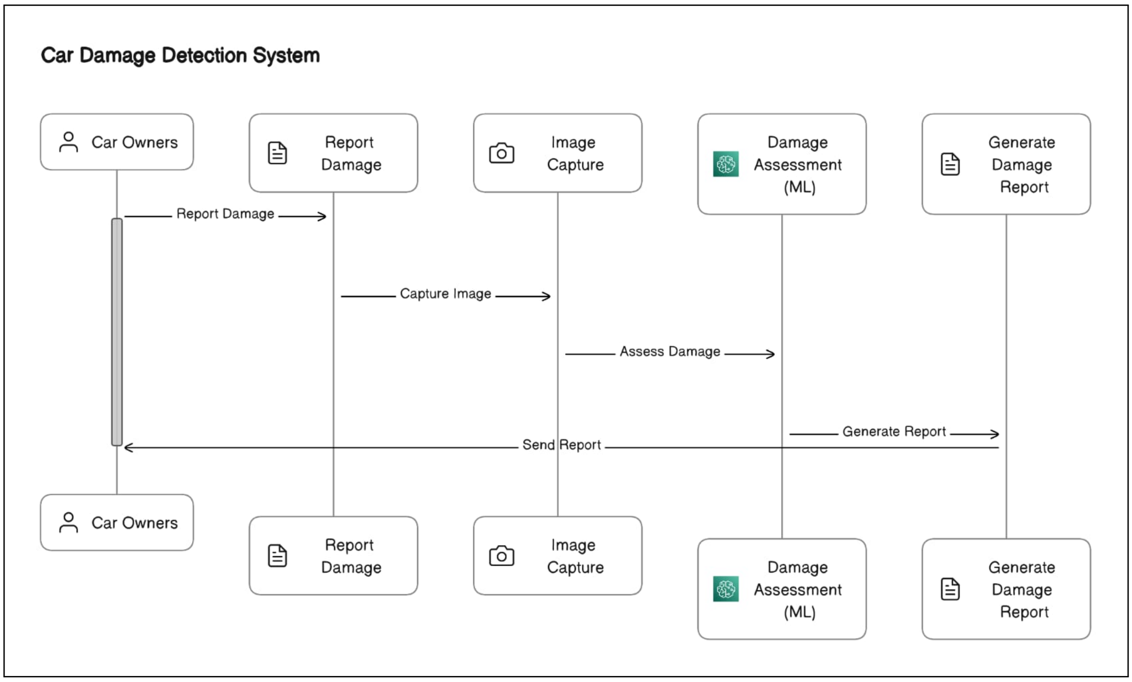

Figure 6.4: Sequence Diagram

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.

6.3.5 Component diagram

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.

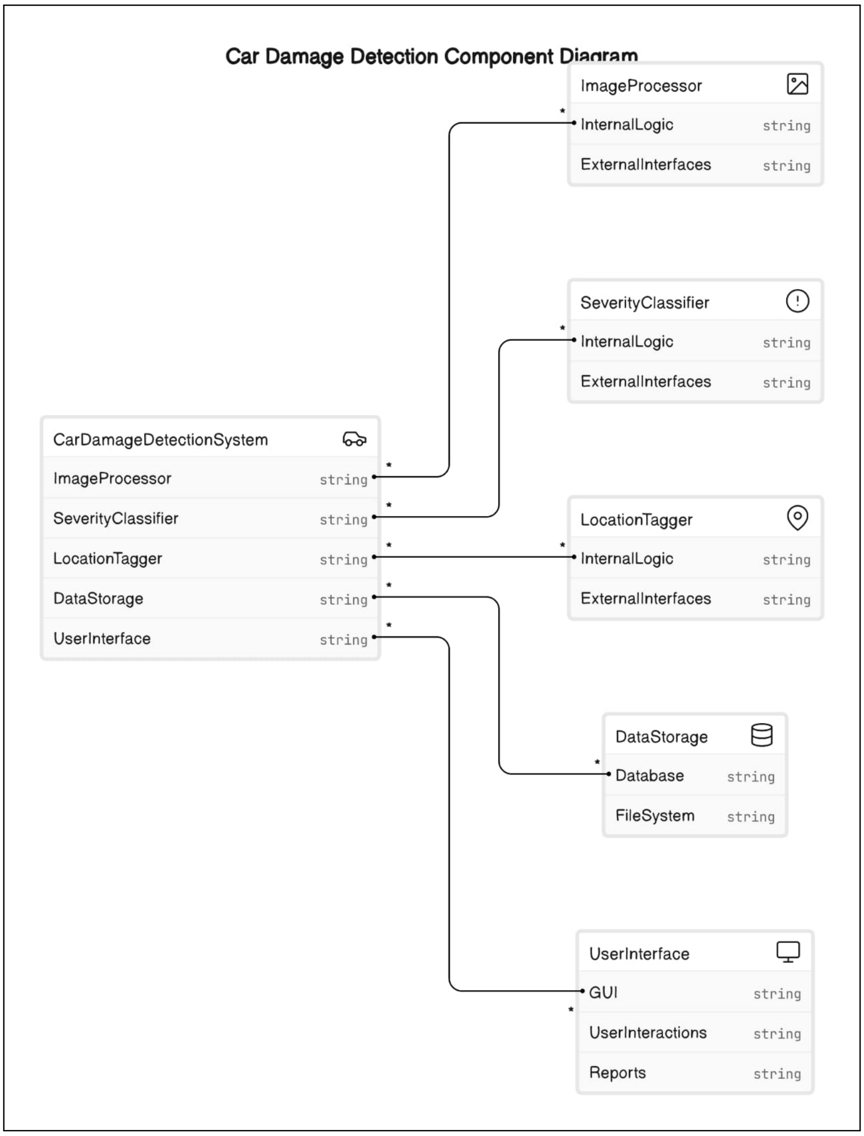

Figure 6.5: Component Diagram

6.3.6 State Transition Diagram

A State Transition Diagram, also known as a State Machine Diagram, is a visual representation used in software engineering and systems design to depict the dynamic behavior of a system. It provides a structured way to illustrate how an object or system can exist in various states and how it transitions between these states in response to specific events or conditions. Each state is represented as a labeled rectangle with rounded corners, defining conditions or situations the system can be in, such as ”Idle” or ”Active.” Transitions, depicted as arrows, specify how the system moves from one state to another, with events or conditions, like ”Start” or ”Error Detected,” triggering these transitions. Events, originating from external sources,

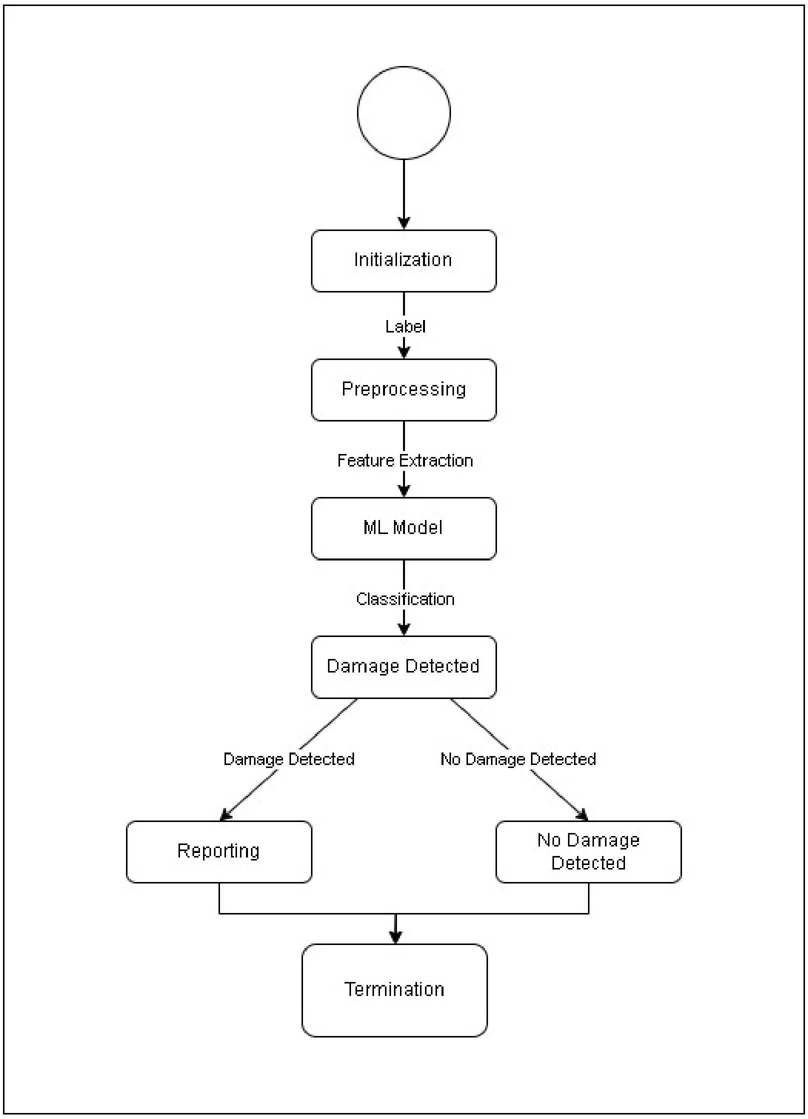

Figure 6.6: State Transition Diagram

determine when and how state changes occur. Actions associated with transitions specify the operations that take place when the system changes state. The initial state, marked with a filled circle, indicates where the system begins, while the final state, represented by a circle with a dot, signifies the end of the state transition process. State transition diagrams are invaluable tools for modeling and understanding complex system behavior, facilitating early issue identification, and contributing to the development of efficient and robust software systems.

CHAPTER 7

DETAILED DESIGN DOCUMENT USING APPENDIX A AND B

7.1 INTRODUCTION

7.2 ARCHITECTURAL DESIGN

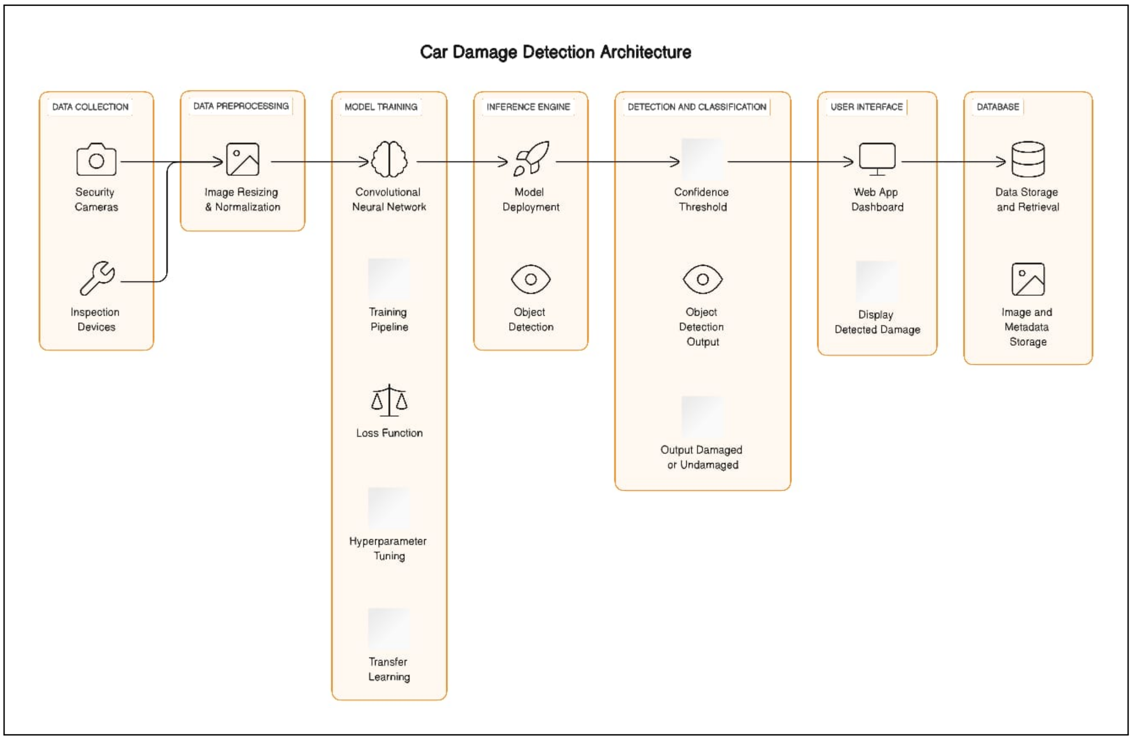

Figure 7.1: Architectural Diagram

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. It enables efficient image processing, classification, and tagging while providing users with a seamless and responsive experience. The physical deployment ensures accessibility from various devices, making it a versatile solution for stakeholders in the automotive industry.

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.

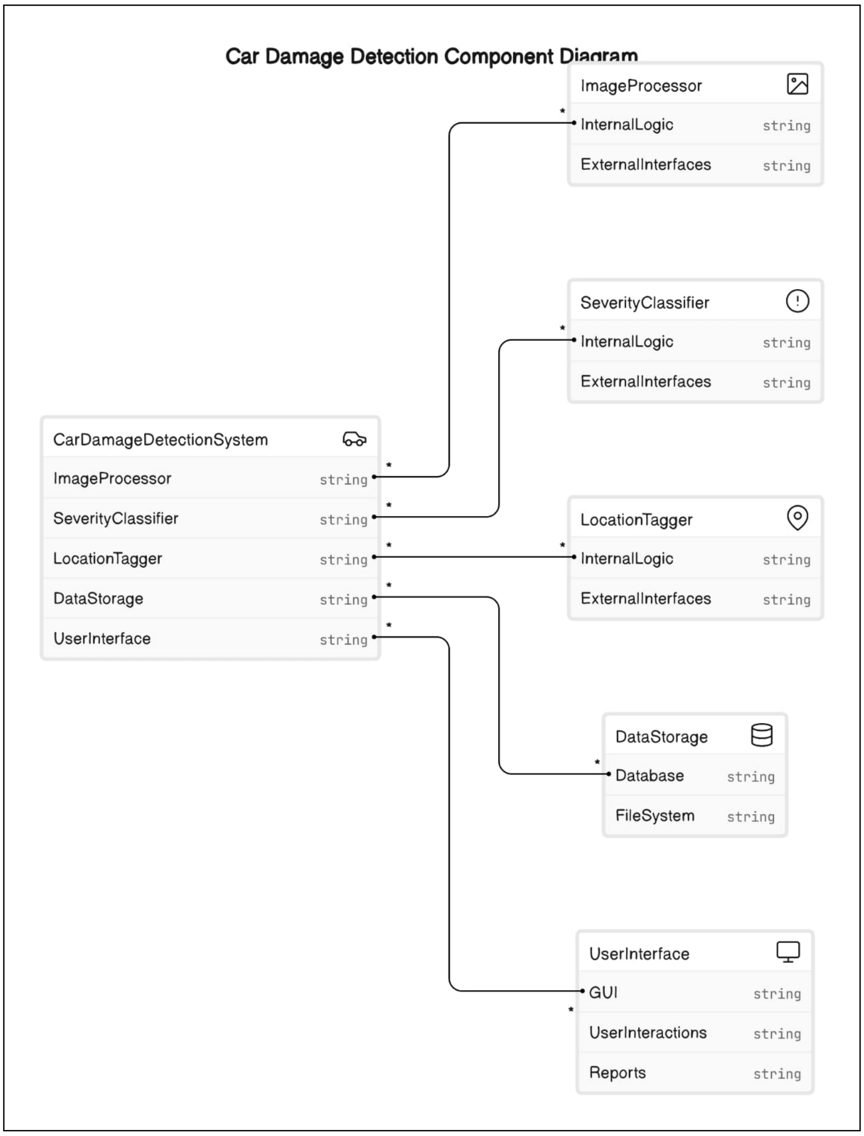

Figure 7.2: Component Diagram

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.

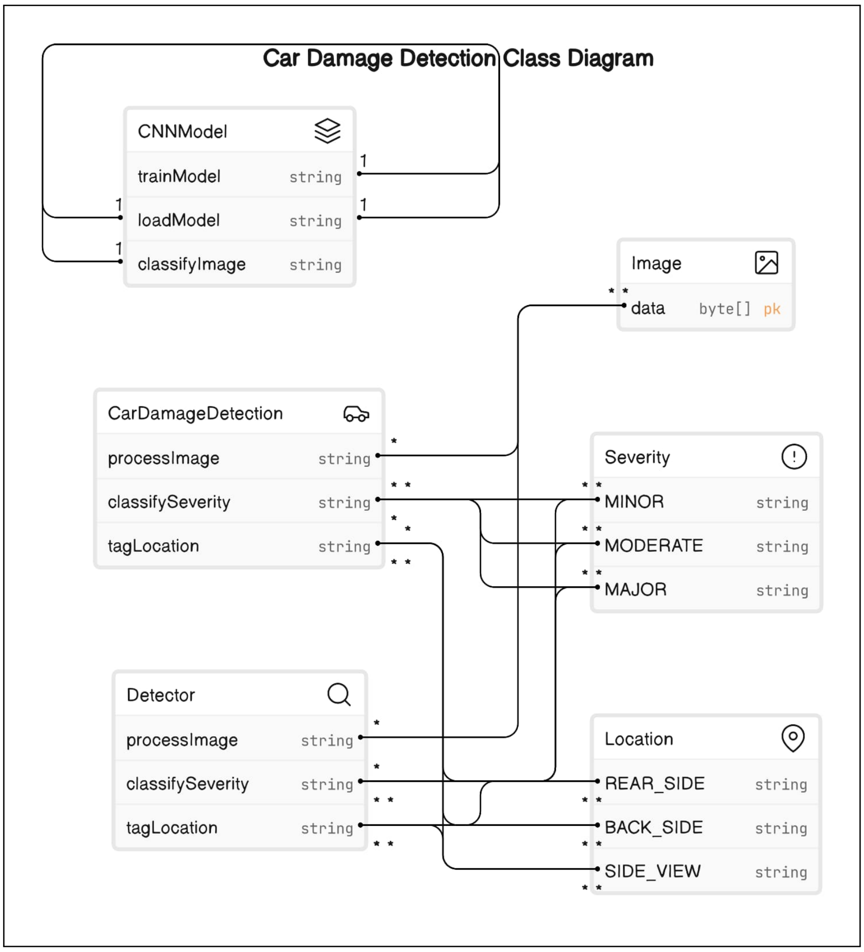

Figure 7.3: Class Diagram

CHAPTER 8

SUMMARY AND CONCLUSION

The Car Damage Detection project represents a significant endeavor in the field of computer vision and machine learning. This project aimed to develop an advanced system for automating the assessment of car damage based on images, enhancing the efficiency and accuracy of this critical process.The project's objectives included the design and implementation of a comprehensive car damage detection system. The system's components were meticulously planned, including image processing, severity classification, location tagging, data storage, and a user-friendly interface. A Convolutional Neural Network (CNN) model was integrated for image analysis, and enums for severity and location were defined.The team successfully implemented the core functionalities, allowing users to upload images for assessment. The system then processed the images, classified damage severity, and tagged the damage location. The results were stored in a well-organized database, ensuring efficient data management.

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 user interface, automating tasks like 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 completion of this project lays the foundation for further enhancements and real-world applications. 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 a valuable contribution to the field of computer vision and sets the stage for future innovations in automated image-based assessments.

CHAPTER 9

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. Wararatpanya, “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 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 Pages 1–6 https://doi.org/10.1145/3406601.3406651.

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

8 Sourish Dey .2019. “CNN Application-Detecting Car Exterior Damage”.

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

It should include assignments such as

- Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading
- Use of above to draw functional dependency graphs and relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.
- Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability. Write also test cases [Black box testing] for each identified functions. You can use Mathematica or equivalent open source tool for generating test data.
- Additional assignments by the guide. If project type as Entreprenaur, Refer [**?**],[**?**],[**?**], [**?**]

ANNEXURE C

PROJECT PLANNER

Using planner or alike project management tool.

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)

1. Paper Title:
2. Name of the Conference/Journal where paper submitted :
3. Paper accepted/rejected :
4. Review comments by reviewer :
5. Corrective actions if any :

ANNEXURE E

PLAGIARISM REPORT

Plagiarism report