DAYANANDA SAGAR UNIVERSITY

Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru-562112

SCHOOL OF ENGINEERING

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

A Project Report On

Hybrid SAM-ResNet Model and YOLOV8 for Car Defect Detection

By

ELURI NAGA YASWANTH KUMAR -ENG21AM0036
G.SURYA ABHIRAM - ENG21AM0039
JEEVIKA.M - ENG21AM0051
SAI BHAVYA SREE.N - ENG21AM0105

Under the supervision of
Dr. VINUTHA N
Associate Professor
Computer Science & Engineering (AI & ML)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY

(2024 – 2025)

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

Department of Computer Science & Engineering

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru – 562112

Karnataka, India

CERTIFICATE

This is to certify that the project entitled **"Hybrid SAM-ResNet Model and YOLOV8 for Car Defect Detection"** is carried out by **ELURI NAGA YASWANTH KUMAR (ENG21AM0036), G. SURYA ABHIRAM (ENG21AM0039), JEEVIKA.M (ENG21AM0051), SAI BHAVYA SREE.N (ENG21AM0105)**, bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore, in partial fulfillment for the award of a degree in Bachelor of Technology in Computer Science and Engineering, during the year **2024 - 2025**.

Dr. Vinutha N	**Dr. Vinutha N**	**Dr. Jayavrinda Vrindavanam**
Associate Professor	Project Co-ordinator	Professor & Chairperson
Dept. of CSE (AIML)	Dept. of CSE (AIML)	Dept. of CSE (AIML)
School of Engineering	School of Engineering	School of Engineering
Dayananda Sagar University	Dayananda Sagar University	Dayananda Sagar University
Signature	Signature	Signature ....

Name of the Examiners:	Signature with date:
1...........................	....
2............................	....
3............................	....

DECLARATION

We, **ELURI NAGA YASWANTH KUMAR (ENG21AM0036), G. SURYA ABHIRAM (ENG21AM0039), JEEVIKA M. (ENG21AM0051), SAI BHAVYA SREE N. (ENG21AM0105)**, are students of the eighth semester B.Tech in Computer Science and Engineering (AI & ML) at the School of Engineering, Dayananda Sagar University. We hereby declare that the Major Project titled **"Hybrid SAM-ResNet Model and YOLOV8 for Car Defect Detection"** has been carried out by us and submitted in partial fulfillment for the award of a degree in **Bachelor of Technology** in **Computer Science and Engineering** during the academic year **2024–2025**.

Student: **Signature**

Name 1: ELURI NAGA YASWANTH KUMAR
USN: ENG21AM0036

Name 2: G.SURYA ABHIRAM
USN: ENG21AM0039

Name 3: JEEVIKA.M
USN: ENG21AM0051

Name 4: SAI BHAVYA SREE.N
USN: ENG21AM0105

Place: Bangalore
Date:

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work. First, we take this opportunity to express our sincere gratitude to School of Engineering& Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Jayavrinda Vrindavanam, Department Chairperson, Computer Science and Engineering (Artificial Intelligence and Machine Learning), Dayananda Sagar University**, for providing right academic guidance that made our task possible.

We would like to thank our guide **Dr.Vinutha N, Associate Professor, Dept. of Computer Science and Engineering of Artificial Intelligence and Machine Learning Dayananda Sagar University**, for sparing her valuable time to extend help in every step of our project work, which paved the way for smoothprogress and fruitful culmination of the project.

We would like to thank our **Project Coordinator Dr. Vinutha N** as well as all the staff members of Computer Science and Engineering (Artificial Intelligence and Machine Learning) for their support. We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Detecting car dents and scratches plays a vital role in automating vehicle damage assessment, particularly in the insurance and automotive sectors. Manual inspection is often time-consuming, subjective, and prone to human error. To address this, the project explores a comparative analysis of two cutting-edge deep learning approaches: YOLOv8 and SAM (Segment Anything Model) integrated with ResNet50.YOLOv8 is a modern object detection model known for its speed and accuracy. It offers real-time detection with high precision, making it suitable for deployment in live inspection systems. Its ability to generate tight and accurate bounding boxes around damaged areas allows for efficient identification of dents and scratches, even under varying lighting and background conditions. On the other hand, the combination of SAM and ResNet50 leverages powerful segmentation techniques and deep feature extraction. While ResNet50 captures high-level visual features, SAM is capable of producing fine-grained segmentation masks that help localize even subtle damage patterns. This makes the SAM+ResNet50 pipeline particularly effective in identifying small, shallow scratches or minor dents that might be missed by standard detection methods.

The dataset used in this study consists of annotated car images featuring a diverse range of damage types, angles, and lighting conditions. Both models are trained and tested using evaluation metrics such as precision, recall, and mean Average Precision (mAP). YOLOv8 is evaluated based on its speed and detection efficiency, whereas SAM+ResNet50 is assessed for its segmentation accuracy and ability to distinguish intricate damage details.This comparative study highlights the strengths of both models for automated damage detection.

Chapter 1

INTRODUCTION

In recent years, the demand for automation in various industrial sectors has grown significantly, especially within the automotive and insurance industries. One area that stands out in this trend is the automated detection of car dents and scratches. Traditional vehicle inspection methods often rely on manual assessments, which are susceptible to human error, inconsistencies, and delays. As a result, there is a pressing need for a more reliable, efficient, and consistent approach to detecting external damages in vehicles.

This project addresses this challenge by developing a robust system for detecting dents and scratches on cars using computer vision and deep learning techniques. Specifically, it focuses on the implementation and evaluation of two advanced models: YOLOv8 and a hybrid approach combining the Segment Anything Model (SAM) with ResNet50. The primary objective of the project is to explore, compare, and analyze the performance of these two techniques in accurately identifying and classifying surface-level damages on vehicles.

The YOLOv8 model is a recent addition to the YOLO (You Only Look Once) family, which has gained popularity for its real-time object detection capabilities. YOLOv8 features an optimized architecture that ensures a balance between speed and accuracy, making it a strong candidate for tasks that demand immediate and precise results. Its design enables it to detect and localize objects efficiently, even under challenging conditions such as poor lighting or complex car surfaces. This makes it particularly suitable for applications where quick, on-the-fly damage detection is essential.

On the other hand, the SAM + ResNet50 hybrid model presents a different strategy. The Segment Anything Model (SAM), developed by Meta AI, is a powerful tool for segmentation tasks, capable of identifying and isolating specific regions of interest within an image. When paired with ResNet50, a deep convolutional neural network well-known for its feature extraction and classification capabilities, the resulting hybrid model offers a comprehensive solution that not only highlights the exact damaged areas but also classifies them effectively. This approach is particularly advantageous when dealing with subtle damages, such as micro-scratches or overlapping dents, that may not be easily distinguishable by traditional object detection models.

By conducting this comparative study, the project contributes valuable insights into the strengths and limitations of detection-based versus segmentation-based approaches. It also provides a foundation for further research and development in automated vehicle damage detection, paving the way for more intelligent, scalable, and accessible solutions in the automotive ecosystem.

The YOLOv8 model, part of the YOLO (You Only Look Once) family, is well-known for its real-time object detection capabilities. It employs an optimized architecture that balances speed and accuracy, making it suitable for detecting dents and scratches under varied lighting conditions and car surfaces. The model leverages its ability to localize objects within images and classify them accurately, providing precise detection results.

On the other hand, the SAM + ResNet50 approach offers a different perspective by integrating SAM, a state-of-the-art segmentation model, with ResNet50, a powerful convolutional neural network. This hybrid model is designed to segment specific regions of interest, such as dents and scratches, followed by classification using ResNet50. This combination enhances the model's ability to isolate minor details and subtle damages that might be challenging to detect with traditional object detection methods.

This project conducts a comprehensive evaluation of YOLOv8 and SAM integrated with ResNet50 by employing a variety of quantitative and qualitative metrics. The comparison includes accuracy, precision, recall, and computational efficiency, offering a clear picture of each model's performance. Beyond numerical evaluation, the study also considers real-world challenges such as noisy backgrounds, reflections, and overlapping damage regions, which often complicate damage detection tasks.

By analyzing how each model responds to these complexities, the project provides a well-rounded assessment of their robustness and reliability. Ultimately, this comparison aims to identify the most effective approach for practical deployment in automotive damage assessment systems, offering valuable insights for use in insurance claims, vehicle inspections, and automated appraisal processes.

1.1 Scope

The scope of this project encompasses the design, implementation, and evaluation of an automated system for detecting car dents and scratches using deep learning-based computer vision techniques. It primarily focuses on comparing two state-of-the-art methods: YOLOv8, a fast and accurate object detection model, and a hybrid model that combines the Segment Anything Model (SAM) with ResNet50, which emphasizes detailed segmentation and classification. Both models are trained and tested on a carefully curated dataset consisting of vehicle images with varying types of external damages. The project aims to assess the strengths and limitations of each model in detecting both prominent and subtle damages under diverse conditions.

This comparison includes evaluating how well each model performs in terms of accuracy, precision, recall, F1-score, and computational efficiency. Special attention is given to understanding how these models behave in practical scenarios, such as images taken under poor lighting, with reflections, or containing noisy backgrounds. The models are expected to not only locate the damaged regions but also differentiate between dents and scratches with a high level of reliability.

Furthermore, the scope includes analyzing how these techniques can be integrated into real-world automotive workflows. Automated damage detection has the potential to greatly assist various stakeholders, such as insurance companies, by speeding up claim assessments and reducing manual verification overhead. Similarly, service centers and vehicle owners can benefit from consistent and accurate evaluations of damage severity, enabling better planning for repairs and maintenance. The results from this study can also contribute to the development of intelligent vehicle inspection systems, which could be embedded in autonomous or semi-autonomous vehicle platforms to ensure ongoing self-assessment of vehicle condition.

Overall, this project not only investigates technical performance but also envisions the practical implications and industrial applications of using AI-driven methods for car damage detection. It sets the foundation for further research and development in the domain of automated visual inspection systems, encouraging future innovation in automotive damage assessment through scalable and intelligent solutions.

Chapter 2

PROBLEM DEFINITION AND OBJECTIVE

2.1 Problem Definition:

Manual inspection of vehicle dents and scratches is time-consuming, inconsistent, and dependent on the inspector's experience and environmental factors. These limitations often lead to delays and inaccuracies in insurance claims and repair decisions. With the rise of AI and computer vision, there's a growing shift toward automated damage detection systems for faster and more reliable results. This project compares YOLOv8 and a hybrid SAM-ResNet50 model to evaluate their effectiveness in detecting and localizing damage. The goal is to identify a solution that offers high accuracy, speed, and robustness for real-world use in the automotive and insurance sectors.

OBJECTIVE

- The main objective of this project is to design and evaluate a reliable and automated solution for detecting dents and scratches on car exteriors using advanced deep learning techniques. Traditional methods for assessing vehicle damage heavily rely on human visual inspection, which not only consumes a significant amount of time but also varies in accuracy due to differences in human judgment, lighting conditions, and inspection angles. These limitations often lead to inconsistent evaluations, delays in processing insurance claims, and an overall increase in operational costs for repair centers and insurance providers.

- With advancements in computer vision and artificial intelligence, there is a growing opportunity to automate damage detection processes using image-based machine learning models. This project specifically focuses on comparing two leading approaches in this space: the You Only Look Once version 8 (YOLOv8) object detection algorithm and a hybrid method co mbining the Segment Anything Model (SAM) with a ResNet50 backbone for feature extraction.

- YOLOv8 is a cutting-edge model known for its real-time object detection capabilities, offering a good balance between speed and accuracy. It is particularly useful in scenarios that demand fast and reliable identification of objects within an image. In contrast, the SAM–ResNet50 model integrates image segmentation with deep feature extraction, allowing for more precise identification and delineation of irregular damage shapes such as surface-level scratches or minor dents, which may be missed by standard object detectors.

- This project involves training and testing both models on a diverse dataset of vehicle images containing visible damages captured under varying environmental conditions. Each model's performance will be analyzed using key metrics such as accuracy, precision, recall, inference speed, and robustness to changes in lighting and camera angle. Furthermore, the quality of the damage localization (how well the model outlines the affected area) will also be considered, especially in use cases where detailed analysis is crucial.

- The ultimate aim is to determine which model or combination of models is best suited for practical applications in the automotive industry. This includes deployment in insurance mobile apps for quick claim assessments, integration with service center diagnostic tools, and usage in automated vehicle inspection systems. By delivering faster, more consistent, and scalable damage detection, this project aims to reduce turnaround times for insurance claims, lower labor costs in service centers, and enhance the overall experience for car owners through more transparent and efficient repair processes. In doing so, it supports the broader shift toward automation and smart technology in transportation and insurance ecosystems.

- In addition to performance evaluation, the project also aims to investigate the adaptability of these models to real-world conditions, such as variations in background, reflections, different vehicle colors, and partial occlusions. These factors often make damage detection more challenging, especially in uncontrolled outdoor environments. Understanding how well each model generalizes to these variations will be crucial for determining their viability in commercial or mobile applications, where controlled imaging conditions cannot always be guaranteed. This aspect of the study ensures that the selected model is not just technically sound but also practical and robust in diverse deployment scenarios.

- Furthermore, the project seeks to lay the groundwork for a scalable damage detection pipeline that can be integrated into existing automotive software platforms. This includes exploring the ease of integration, model size, inference requirements, and compatibility with edge devices such as mobile phones or embedded systems in inspection booths. By doing so, the objective extends beyond model comparison to include a systems-level view of deployment readiness. The broader goal is to contribute to the advancement of smart vehicle assessment systems that combine automation, accuracy, and user-friendliness—ultimately reducing operational costs, expediting workflows, and enhancing service quality for all stakeholders in the automotive and insurance sectors.

2.2 Novelty of Proposed Approach

This project introduces a comparative deep learning framework combining object detection (YOLOv8) and segmentation-based analysis (SAM with ResNet50) for precise car dent and scratch detection. The architecture leverages YOLOv8 for real-time detection and SAM–ResNet50 for detailed damage localization across diverse imaging conditions. systems. Targets conditions like spasticity and motor neuron disorders

- Hybrid Integration of Detection and Segmentation: The project introduces a novel hybrid model by combining Segment Anything Model (SAM) with ResNet50, enabling pixel-level damage segmentation enhanced by deep feature extraction, which is rarely used together for car damage detection.

- Comparative Study with YOLOv8: Unlike many studies that rely solely on a single object detection model, this project presents a comparative evaluation of two state-of-the-art models—YOLOv8 for bounding box detection and SAM+ResNet50 for precise segmentation—offering insights into their relative strengths.

- Pixel-Level Precision with SAM: SAM is utilized to segment dents and scratches at the pixel level, enabling more accurate and detailed localization of even minor or overlapping damages—something bounding boxes alone may miss.

- Enhanced Feature Representation through ResNet50: The integration of ResNet50 improves the model's ability to extract high-level features, which boosts performance in complex visual scenarios such as reflective surfaces or subtle dents.

- Application-Driven Focus: The project is tailored for real-world automotive use-cases such as insurance claim automation, vehicle resale evaluation, and service diagnostics, ensuring practical relevance beyond academic exploration.

- Custom Annotated Dataset: Utilizes a specially curated and annotated dataset of car images with varied damage types, angles, and lighting, providing a robust foundation for training and evaluation.

- Scalability and Flexibility: The proposed framework is modular and can be adapted or extended to other vehicle types or damage categories, showing strong potential for scalability and future enhancement.

Chapter 3

LITERATURE SURVEY

[1] Vijay M, Vijay V, Jegan, B.RAMA.M.E. ”CAR DAMAGE DETECTION USING MACHINE LEARNING ”. IJCRT 5 May 2024

The study addresses challenges like limited dataset diversity and the absence of standardized benchmarks, which complicate car damage classification. Using the YOLOv8 model, a robust methodology was developed involving dataset preprocessing, segmentation, and binary masks for precise damage localization. The system showcased high accuracy and efficiency, promising enhanced applications in insurance, vehicle maintenance, and safety. Future work includes advanced diagnostics and scalability.

[2]Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, Yehia Massoud. ”A Very Deep Transfer Learning Model for Vehicle Damage Detection and Localization”. IEEE 2024

Insurance companies experience significant financial losses due to claims leakage caused by inefficient processes, fraudulent claims, and poor decision management. This research presents a deep learning-based method that utilizes instance segmentation and transfer learning to automate vehicle damage detection and localization, streamlining the claims assessment process. The Inception-ResNetV2 model, employed through transfer learning, has shown better results in feature extraction and damage detection compared to the VGG16 model.

[3]Mohamed A. Abou-Khousa, Mohammed Saif Ur Rahman, Kristen M. Donnell, Mohammad Tayeb Al Qaseer. ”Detection of Surface Cracks in Metals Using Microwave and Millimeter-Wave Nondestructive Testing Techniques—A Review”. IEEE 2023

This study explores advancements in microwave and millimeter-wave nondestructive testing for detecting surface cracks in metals. It highlights far-field, near-field, and resonator methods, emphasizing probe designs and imaging techniques. Challenges include sensitivity at larger lift-offs and detecting covered cracks. These methods offer promise for accurate detection under complex conditions, with scope for further refinements.

[4] Vignesh Sampath, Inaki Maurtua, Juan Jose Aguilar Martin, Aitor Gutierrez. "Attention-Guided Multitask Learning for Surface Defect Identification". IEEE SEPTEMBER 2023

The study proposes Defect-Aux-Net, a multitask learning model for surface defect detection, classification, and segmentation. Using FPN and ResNet-50, it integrates channel and spatial attention mechanisms to enhance defect detection in noisy backgrounds. Key challenges included limited datasets and false positives. Results showed superior performance, achieving 97.1 accuracy, a Dice score of 0.926, and mAP of 0.762, demonstrating its effectiveness in industrial defect identification.

[5]EFSTATHIOS BRANIKAS, PAUL MURRAY GRAEME WEST. "A Novel Data Augmentation Method for Improved Visual Crack Detection Using Generative Adversarial Networks". IEEE February 2023

This study introduces a novel data augmentation technique for visual crack detection using CycleGAN. The methodology involves domain transfer to generate annotated, realistic crack images, enhancing training datasets without manual labeling. Challenges include addressing data imbalances and improving segmentation accuracy. Results demonstrate significant improvements in crack detection performance across diverse datasets, highlighting the approach's potential for efficient defect detection in various industrial applications.

[6] NYOMAN BOGI ADITYA KARNA, SYIFA MALIAH RACHMAWATI, MADE ADI PARAMARTHA PUTRA, MIDETH ABISADO, GABRIEL AVELINO SAMPEDRO. "Toward Accurate Fused Deposition Modeling 3D Printer Fault Detection Using Improved YOLOv8 With Hyperparameter Optimization". IEEE June 2023

This research article presents an enhanced YOLOv8 model with an additional feature extraction layer integrated into the traditional YOLOv8 architecture to improve fault detection performance in smart additive manufacturing, specifically for FDM 3D printers.The challenges include evaluating another model hyperparameter, such as optimizer, learning rate, and epoch,implementation on a larger scale. The best results are achieved using the YOLOv8s model with an image input size of 640 and a batch size of 16, achieving a mAPval(50-95) of 89.7

[7] Ebrahim Khalili, Blanca Priego-Torres, Antonio Leon-Jimenez and Daniel Sanchez-Morillo. "Automatic lung segmentation in chest X-ray images using SAM with prompts from YOLO" 2022

The study tackles challenges in automated lung segmentation, including dataset bias and variability in lung pathologies. Combining SAM and YOLO models, it optimizes prompts and segmentation accuracy. The robust framework demonstrates improved generalization across datasets. Future work includes diagnostic integration and segmentation refinement for advanced clinical applications, supporting scalable, automated lung analysis.

[8]Atharva Shirode, Aparna Halbe, Tejas Rathod, Parth Wanjari. ”Car Damage Detection and Assessment Using CNN”. IEEE 2022

The study focuses on automating car damage detection to address the limitations of manual verification, such as inefficiency and resource usage. Using VGG16 and Mask R-CNN, the methodology identifies damage presence, location, severity, and masks the affected region. Results demonstrate effective detection despite challenges with image noise. Future work aims to enhance dataset size and integrate repair cost assessment.

[9]Ranjodh Singh, Meghna P Ayyar, Tata Venkata Sri Pavan, Sandeep Gosain, Rajiv Ratn Shah. ”AUTOMATING CAR INSURANCE CLAIMS USING DEEP LEARNING TECHNIQUES”. IEEE 2019

This study proposes an automated car insurance claim system using deep learning. Mask R-CNN and PANet localize car parts and damages, while VGG16 identifies damage severity. Challenges include handling diverse shapes, lighting, and image quality. Results show a mean average precision (mAP) of 0.40 for damage detection, enabling efficient, reliable claims processing and reducing manual inspection efforts.

[10] Kalpesh Patil, Mandar Kulkarni, Anand Sriraman, Shirish Karande. ”Deep Learning Based Car Damage Classification”.16th IEEE 2017

This study focuses on car damage classification using deep learning. A custom dataset was created, and models like CNN, transfer learning, and ensemble learning were implemented. Challenges included limited labeled data and high inter-class similarity. Transfer learning with ensemble methods achieved 89.5 accuracy. Data augmentation and damage localization further improved classification efficiency, supporting automated insurance claims.

Chapter 4

METHODOLOGY

This project involves detecting car dents and scratches using two models, YOLOv8 and SAM+ResNet50. The methodology is structured into key stages: data preparation, model training, evaluation, and result analysis.

4.1 Dataset Preparation

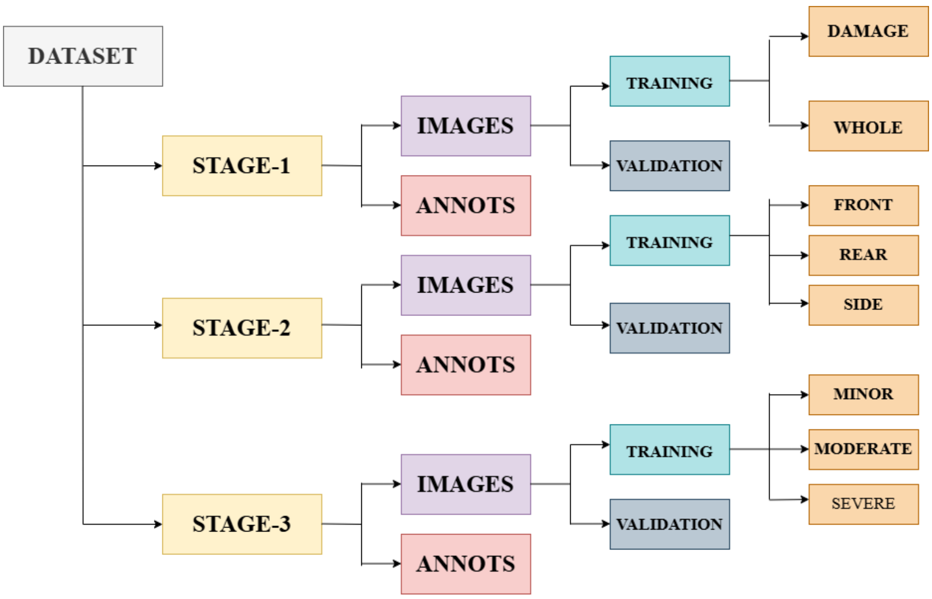

Figure 4.1: Dataset Workflow for Vehicle Damage Classification

Dataset Collection & Preprocessing

The workflow for the dataset follows a well-defined, stepwise method aimed at accurately classifying and evaluating vehicle damage. The process begins with an initial phase focused on separating the dataset into images and their respective annotations. These annotations, which indicate the locations of dents and scratches on the vehicle, are crucial for training the models. After separating the data, it is divided into two primary subsets: training and validation sets. The training set is used to teach the models how to identify the presence of vehicle damage, while the validation set is used to monitor the model's progress and adjust parameters during the training phase.

In this initial stage, the focus is on determining whether a vehicle exhibits any visible damage, essentially performing a binary classification of damaged vs. undamaged vehicles. This step ensures that the models first understand the general presence of damage before moving on to more precise tasks.

Once the initial classification of damage presence is complete, the second phase of the workflow shifts to accurate damage localization. In this phase, the dataset is further refined by organizing images with their corresponding labeled annotations that highlight specific damage regions.

The model is then trained to detect and localize these damaged areas in more detail. For the localization task, the damage is categorized into three main zones of the vehicle: the front, rear, and side sections. This classification of damage zones offers distinct perspectives on where collisions may have occurred, enabling the model to analyze potential impact points with greater precision. By understanding the location of the damage, the model can more accurately assess the severity and type of impact, leading to a better understanding of how collisions may affect the vehicle.

In the final phase, the focus shifts to damage severity evaluation. Using the training procedure developed in the earlier phases, the damage is classified into three distinct levels of severity: minor, moderate, and severe. This tiered classification system allows the model to evaluate the extent of the damage more comprehensively. The minor category includes slight abrasions or small dents, the moderate category captures moderate dents or scratches, and the severe category accounts for deep dents or cracks that may require extensive repair. By categorizing damage in this way, the model can generate more accurate and actionable insights for applications such as insurance claims, vehicle servicing, or autonomous inspection systems. This structured and multi-level classification approach maximizes the computational model's ability to handle a wide range of vehicle damage patterns, improving its overall performance and efficiency.

Together, these phases form a robust data preparation strategy that enables the detection, localization, and classification of vehicle damage in a systematic and hierarchical manner. By separating the workflow into these distinct steps—damage detection, localization by vehicle zones, and severity evaluation—the project ensures that each model can be trained to handle specific aspects of damage analysis in a focused manner, thus optimizing the overall accuracy and reliability of the damage detection system.

4.2 Framework Model Selection:

In the frameworks first approach utilizes YOLOv8, a real-time object detection model renowned for its speed and accuracy in localizing multiple objects within an image. The second approach is a hybrid model that combines the powerful segmentation capabilities of the Segment Anything Model (SAM) with the deep feature extraction strengths of ResNet50. While YOLOv8 focuses on detecting and classifying damage through bounding boxes, the SAM-ResNet50 model enables more precise region-based identification of subtle damages.

4.2.1 YOLOV8

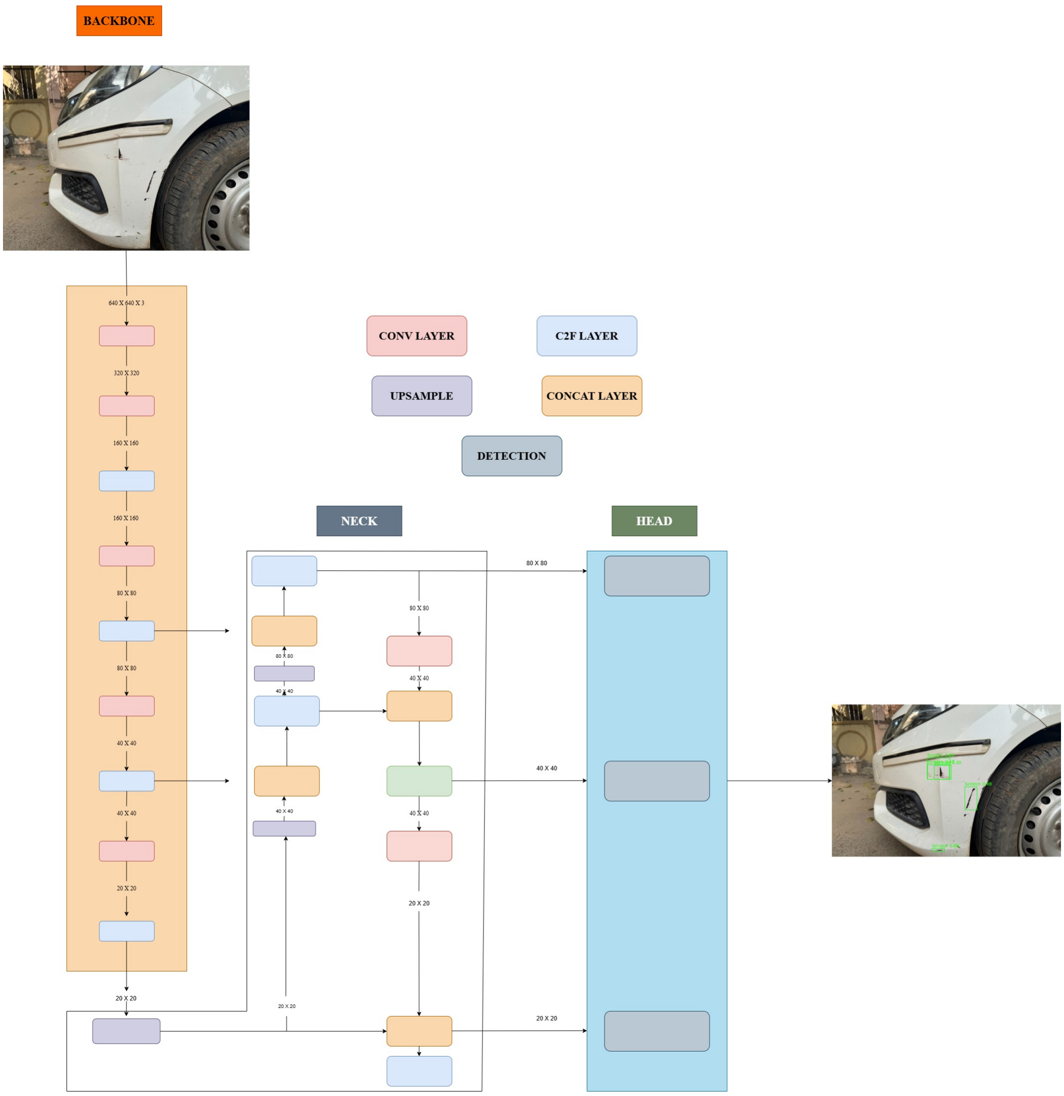

Figure 4.2: YOLOv8 Architecture for Vehicle Damage Detection

1. Input and Preprocessing

The initial stage of the YOLOv8 pipeline starts with the raw images that serve as the primary input. These images can either be uploaded by the user or accessed from an existing dataset that contains images of vehicles. These raw images are critical as they provide the visual data necessary for detecting vehicle damage such as scratches and dents. Before these images are fed into the model, several essential preprocessing steps are carried out to ensure the model can effectively work with the data.

The first preprocessing step involves resizing the images. Since YOLOv8, like most neural networks, requires input images to be of a fixed size, resizing is necessary to ensure uniformity across all images. This step is crucial as it ensures that all images are of compatible dimensions, making them ready for the model. Typically, the input size is standardized, such as resizing images to dimensions like 416x416 or 640x640 pixels, depending on the model's requirements.

Following resizing, the pixel values of the images are normalized. This step is important for improving the model's training efficiency and stability. Normalization involves scaling the pixel values so they fall within a specific range, often between 0 and 1, by dividing each pixel's value by 255. This step helps avoid issues related to varying pixel value ranges across different images, which could otherwise cause instability during the learning process.

Once the images are resized and normalized, a key technique called data augmentation is applied to further enhance the training dataset. Data augmentation techniques such as flipping, rotation, cropping, and scaling are commonly used to artificially expand the dataset. These techniques generate new variations of the original images by altering their orientation or size, which helps the model become more robust and less prone to overfitting. Augmentation is particularly helpful when the available dataset is limited, as it increases the diversity of the training data and allows the model to generalize better to new, unseen images. After completing these preprocessing steps, the images are fully prepared and ready to be passed into the YOLOv8 pipeline, where they will be processed in subsequent stages of feature extraction, detection, and classification.

2. Backbone

Feature Extraction In the YOLOv8 architecture, the Backbone is the first major component responsible for feature extraction. The Backbone's primary function is to convert the input image into a set of meaningful, low-level features that can be used for further processing. This is done using a series of convolutional layers that apply filters to the image in order to detect simple patterns such as edges, textures, and basic shapes. In the case of vehicle damage, the Backbone might detect edges where dents or scratches are present or identify the distinct texture patterns on the surface of the vehicle.

The convolutional layers in the backbone are designed to progressively learn more complex features as the image moves through each layer. The first layers focus on detecting simple features, such as lines and corners, while the deeper layers of the backbone begin to capture more abstract patterns, such as curves and detailed structures that might indicate a dent or scratch. These deep features are what allow the model to move beyond simple edge detection and identify higher-level information, like the location and shape of a dent or scratch on the vehicle's body.

The feature extraction process in the backbone is key to making the model effective for detecting specific types of damage, as it helps the model distinguish between different damage patterns and understand the underlying structure of the vehicle. The features that are extracted are passed on to the next stage of the pipeline, the Neck, where they will be further refined and processed. The backbone's role is essential because it forms the foundation of the object detection process, ensuring that the model has the necessary information to make accurate predictions in later stages of the pipeline.

3. Neck

The Neck plays a critical role as the intermediary between the Backbone and the Head in the YOLOv8 architecture. While the Backbone extracts the raw features, the Neck's job is to enhance these features, making them more useful for object detection. It does this by applying additional convolutional layers to the feature maps provided by the Backbone. The Neck aims to emphasize the most important regions of the image, refining the feature maps to highlight areas where the model is more likely to find damage, such as the front, rear, or side of the vehicle.

The enhancement process in the Neck allows the model to focus on key regions that are most likely to contain dents or scratches.If the model detects a large, smooth surface in the middle of the vehicle, the Neck will focus more attention on the edges or corners where damage is more likely to occur. By enhancing these key areas, the Neck helps the model become more efficient and accurate, ensuring that it doesn't waste computational resources on irrelevant parts of the image.

The fully connected layer within the Neck is responsible for aggregating and organizing the enhanced features, preparing them for the final detection phase. This stage ensures that the features are properly structured and ready to be processed by the Head, which will make the final decision on what objects (or damages) are present in the image. The Neck is vital for refining the feature representations in ways that improve the model's ability to detect subtle or hard-to-spot damage, such as small scratches that might otherwise go unnoticed. In this way, the Neck plays an essential role in enhancing the model's detection capabilities.

4. Head

The Head is the final stage of the YOLOv8 pipeline, responsible for taking the refined features from the Neck and performing object detection. Object detection involves identifying and localizing objects (in this case, vehicle damage) within an image. The Head utilizes pre-trained models and complex algorithms to determine the bounding boxes of objects, classify them, and assign a confidence score that reflects how likely it is that the object belongs to a particular class.

Bounding boxes are rectangular areas that the model draws around detected objects, indicating their location in the image. These boxes are crucial for defining the precise position of vehicle damage, such as where a scratch or dent occurs. In addition to the bounding boxes, the Head also classifies each object within the box, providing a class label such as "scratch" or "dent." This label helps the model distinguish between different types of damage. The confidence score is a numerical value that represents the model's certainty about its prediction. A higher score indicates greater confidence that the object in the bounding box is indeed a dent or scratch, while a lower score suggests more uncertainty.

One of the standout features of the YOLOv8 Head is its ability to process multiple objects simultaneously in a single image. This makes YOLOv8 highly efficient for real-time object detection tasks, where multiple areas of damage may be present on the vehicle. The Head is optimized for speed and accuracy, ensuring that the model can detect damage quickly while maintaining high levels of precision. This capability is particularly valuable in practical applications, such as automated vehicle inspections, where timely and accurate detection of vehicle damage is crucial. By providing both the location and the classification of each damage type, the Head ensures that the YOLOv8 model can effectively handle complex detection scenarios.

4.2.2 SAM+RESNET-50

The proposed methodology outlines a powerful and structured approach for image analysis by combining Convolutional Neural Networks (CNNs) with self-attention mechanisms. This technique is especially effective for tasks such as object detection, image segmentation, and classification, where both detailed visual information and broader image context are important.

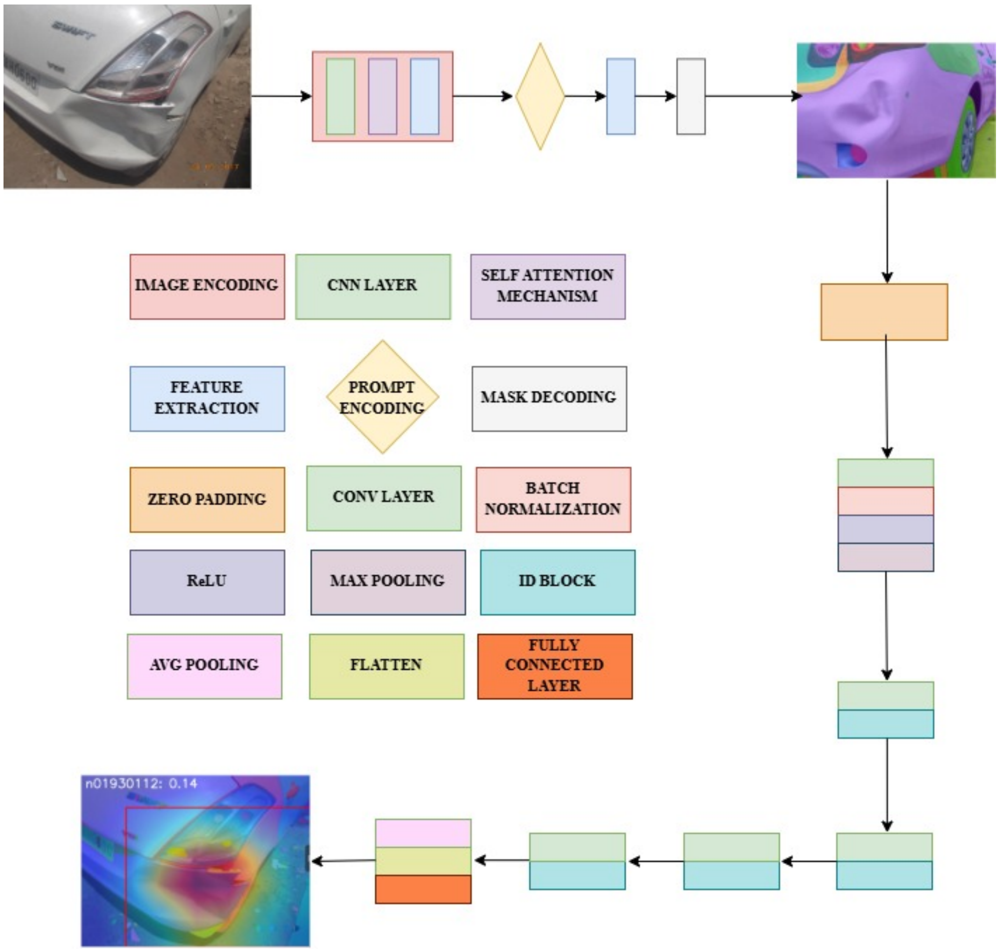

Figure 4.3: SAM + ResNet-50 Architecture for Vehicle Damage Detection

The process begins by passing the input image—such as a photo of a car with visible dents or scratches—through multiple CNN layers. These layers gradually extract features from the image, starting with simple patterns like lines, edges, and textures. As the image progresses deeper through the CNN, more complex features are captured, such as the shape of damaged parts, object outlines, or textures specific to scratches and dents. CNNs are excellent at understanding local spatial structures,which makes them ideal for recognizing patterns that appear in different parts of the image.

Convolutional Neural Networks (CNNs) may struggle to capture long-range dependencies or relationships between distant regions within an image. For instance, if damage is spread across various parts of a car, traditional CNNs might not fully connect these distant regions. To overcome this issue, a self-attention mechanism is added. This component helps the system understand which parts of the image are most important by assigning more focus to certain regions. It enables the model to view the image as a whole and identify relationships across distant areas. This is especially useful in scenarios like huge damage detection without being distracted by irrelevant background elements. The self-attention mechanism improves the model's ability to highlight critical parts of the image and understand the broader context of the scene, leading to more accurate and informed predictions.

Once the image has been processed and important features are extracted, these features are sent through a prompt encoding module. This module reshapes and converts the data into a suitable form for the next stages of the pipeline. It ensures the extracted features are properly aligned and formatted for advanced processing. These refined features are then passed into another feature enhancement module, which sharpens and strengthens the feature representation. This additional processing helps make sure the features are clear and strong enough for the upcoming prediction tasks.

Following this, the refined features go through a segmentation stage where a mask is created. This mask highlights specific regions of interest in the image for example, the areas on a car that are damaged. The mask gives the model a clear view of the relevant parts of the image that need further analysis. Sometimes, the size or shape of the output mask does not match the required format. To solve this, zero-padding is applied to adjust the mask dimensions and ensure consistency throughout the model. This step helps maintain uniform input shapes during model execution.

After segmentation and padding, the image data passes through several essential operations. These include convolutional layers for pattern detection, batch normalization to stabilize data flow, ReLU activation for introducing non-linearity, and max pooling to reduce the size of the data while keeping important information. These operations work together to filter, transform, and compress the data without losing critical details. Then, the output is processed through multiple conv blocks and identity blocks. These blocks are connected with residual paths, which allow the original features to be carried forward to deeper layers. This design improves information flow and prevents data from being lost as it moves through the network. It also makes training more stable and efficient by reducing the chance of issues like vanishing gradients.

In the final stage of the pipeline, the features that have been extracted and processed are passed through an average pooling layer. This step reduces the spatial size of the feature maps by computing average values, helping to summarize all the important visual information in a compact form. The resulting feature maps are then flattened into a single vector and passed through one or more fully connected layers. These layers produce the final output of the model, which may include predictions such as the type of damage, its severity level, or the specific part of the car affected. The system can also classify objects, assign labels, or generate confidence scores based on the application requirements.

By combining CNNs, which are strong in capturing local patterns, with self-attention mechanisms, which offer a global view of the image, this method provides a balanced and robust solution for image understanding. The integration of these two techniques ensures that the model can detect small, subtle issues while still keeping the overall structure and context of the image in mind. This approach is especially beneficial for damage detection tasks, where both precision and contextual awareness are necessary for accurate and reliable results.

4.3 Model Training and Overview:

YOLOV8:

YOLOv8 is a powerful and modern object detection model that has gained significant attention for its impressive performance in real-time tasks. In this context, it is used to detect and locate visible damage on vehicles, such as dents and scratches, by drawing precise bounding boxes around the affected areas. Built on a robust convolutional neural network (CNN) backbone, YOLOv8 excels at identifying objects within images quickly and accurately. The model is initially trained on large, generic datasets that contain diverse visual patterns. These pre-trained weights help the model understand fundamental image structures like edges, curves, and shapes. To tailor it for vehicle damage detection, YOLOv8 is fine-tuned using a custom dataset that contains annotated images of cars showing various types of surface damage. This fine-tuning process helps the model adapt to specific features associated with damaged vehicle surfaces—such as irregular reflections, distortions in the car body, or surface abrasions—which would typically not be present in standard object categories.

The fine-tuning process involves feeding the model with numerous images containing labeled damage areas so it can learn to associate certain visual cues with specific damage types. Over time, YOLOv8 becomes capable of detecting even subtle indicators of damage, such as fine scratches that might be difficult to spot with the naked eye. The model not only localizes the damaged regions using bounding boxes but also classifies them into predefined categories (such as dent or scratch), allowing for a clear and structured understanding of the vehicle's condition. Its ability to detect multiple damages in a single image, even under varied lighting or complex backgrounds, makes it ideal for real-world applications like insurance assessments, automated repair estimations, or pre-owned car inspections.

SAM+ResNet50:

Segment Anything Model (SAM) is a vision model designed to perform high-precision image segmentation by identifying and outlining objects at the pixel level. SAM delivers fine-grained segmentation by precisely outlining object contours, enabling a clearer understanding of spatial features and supporting detailed analysis of specific regions in an image. In this system, SAM is used to capture the exact shape and boundaries of damaged areas, providing more accurate localization than coarse rectangular approximations. It highlights potential regions with surface anomalies, helping to assess the extent of damage and setting up the image for more refined classification in the following stage.

Once the image regions are segmented using SAM, they are passed through ResNet50 for classification. ResNet50 is a deep CNN model known for its efficiency in extracting detailed features from input data. It uses residual connections to overcome the vanishing gradient problem, allowing it to retain valuable information from earlier layers while processing deeper patterns in the image. These capabilities are particularly important when distinguishing between different types of vehicle damage, as the visual differences between a dent and a scratch can be subtle. Dents often involve depth and distortion, while scratches tend to be more superficial and linear. ResNet50 learns to differentiate these characteristics through repeated exposure to labeled training data.

Once trained, it can accurately classify segmented regions as either dents or scratches based on their visual features.By combining the strengths of YOLOv8, SAM, and ResNet50, this approach creates a robust pipeline for vehicle damage analysis. YOLOv8 ensures efficient and fast detection of damage across the entire image, SAM enhances the precision by outlining the exact areas of interest, and ResNet50 delivers an accurate classification of the type of damage.

This integrated system supports not just visual detection but a complete analysis of the damage, providing valuable insights for downstream applications such as automated report generation, vehicle assessment in service centers, and even integration with mobile apps for user-friendly diagnostics. The combination of detection, segmentation, and classification allows for a thorough and reliable inspection process that minimizes human error and speeds up the evaluation workflow.

4.4 Evaluation Metrics

In this project, various evaluation metrics are employed to assess the performance of the two models—YOLOv8 and SAM+ResNet50—for the task of car dent and scratch detection. These metrics help in quantitatively measuring the accuracy, precision, and overall effectiveness of the models under different conditions. The metrics used include Accuracy, Precision, Recall, F1-Score, Intersection over Union (IoU), Mean Average Precision (mAP), and Inference Time. Below is a detailed description of each metric along with its corresponding formula:

4.4.1 Accuracy

Definition: Accuracy measures the overall correctness of the model by calculating the proportion of correctly predicted instances (both positive and negative) out of the total predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

True Positive (TP): The count of positive cases correctly identified as positive by the model.

True Negative (TN): The count of negative cases correctly identified as negative by the model.

False Positive (FP): The count of negative cases incorrectly identified as positive by the model.

False Negative (FN): The count of positive cases incorrectly identified as negative by the model.

4.4.2 Precision

Definition: Precision measures the proportion of positive predictions that were actually correct. It is especially useful when the cost of false positives is high.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where:

True Positive (TP): The number of instances that are actually positive and are correctly identified as positive by the model.

False Positive (FP): The number of instances that are actually negative but are incorrectly classified as positive by the model.

4.4.3 Recall

Definition: Recall measures the proportion of actual positive instances that were correctly identified. It's important when missing a positive prediction has a high cost.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where:

True Positive (TP): The number of instances that are actually positive and are correctly identified as positive by the model.

False Negative (FN): The number of instances that are actually positive but are incorrectly classified as negative by the model.

4.4.4 F1-Score

Definition: The F1-score is the harmonic mean of precision and recall. It provides a balance between the two, especially when there is an uneven class distribution.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

Precision: The proportion of positive predictions that are actually correct, reflecting how accurate the positive predictions are.

Recall: The proportion of actual positive instances that are correctly identified, indicating how well the model identifies positive cases.

4.4.5 Intersection over Union (IoU)

Definition: IoU is used to evaluate how well the predicted bounding box or segmentation mask overlaps with the ground truth. It is a core metric in both object detection and segmentation tasks.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{A \cap B}{A \cup B}$$

Where:

A = Predicted bounding box or mask

B = Ground truth bounding box or mask

4.4.6 Mean Average Precision (mAP)

Definition: mAP is a comprehensive metric often used in object detection tasks to evaluate the precision-recall trade-off across multiple classes and thresholds.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Where:

$\mathbf{AP}_i$ is the average precision for class i

N is the total number of classes

4.4.7 Inference Time

Definition: Inference time is the average time taken by the model to process a single image and produce output. It reflects the speed and computational efficiency of the model.

$$\text{Inference Time} = \frac{\text{Total Time for Predictions}}{\text{Number of Images}}$$

This metric is typically measured in milliseconds (ms) or seconds per image.

Chapter 5

REQUIREMENTS

5.1 Functional Requirements for YOLOv8 Model Execution

1. **CPU Specifications:** To ensure smooth and uninterrupted operation of the YOLOv8 model, a powerful processor is necessary. A quad-core CPU, preferably an Intel Core i5 (9th generation or newer) or AMD Ryzen 5 (3rd generation or newer), with a base clock speed of at least 2.5 GHz, is required. This level of processing power is critical for managing model operations, preprocessing tasks such as image loading, resizing, augmentation, and for orchestrating data flow through various stages of the model pipeline. The CPU must also be capable of supporting multi-threaded operations for optimized task execution.

2. **GPU Specifications:** A dedicated graphics processing unit (GPU) is essential for executing computationally intensive processes, especially during model training and inference. A minimum of 4 GB of dedicated video RAM (VRAM) is required, with supported models including NVIDIA GeForce GTX 1050 Ti, GTX 1650, or higher. For faster processing and larger batch sizes, a more advanced GPU like the RTX 3060 or 3070 is recommended. The GPU should support CUDA (Compute Unified Device Architecture) and be compatible with the PyTorch and Ultralytics YOLOv8 framework for accelerated tensor operations.

3. **Memory Requirements:** A system equipped with at least 8 GB of RAM is necessary to effectively handle image loading, batch processing, model weight loading, and intermediate feature storage. For larger datasets or more complex model configurations, 16 GB of RAM is recommended to prevent memory bottlenecks and to ensure a seamless training and inference workflow without crashes or slowdowns.

4. **Software Environment:** To support YOLOv8, the following software stack is required:

    - **Operating System:** Windows 10/11, Ubuntu 20.04 or later, or macOS (for CPU-based execution).

- **Python Version:** Python 3.8 or above must be installed to ensure compatibility with the latest versions of the YOLOv8 dependencies.
- **Libraries and Frameworks:**
  - PyTorch (compatible with installed CUDA version)
  - Ultralytics YOLOv8 package (via pip install)
  - OpenCV for computer vision-related image manipulation
  - NumPy for numerical operations and tensor manipulations
  - Matplotlib and Seaborn for optional visualizations and debugging
- **CUDA Toolkit and cuDNN:** Required if using GPU acceleration. The correct version of CUDA (e.g., 11.7) and cuDNN must match the PyTorch installation.

5.2 Functional Requirements for SAM + ResNet-50 Execution

1. **CPU Specifications:** For effective execution of the SAM (Segment Anything Model) and ResNet-50, a powerful multi-core CPU is required. An Intel Core i5 or AMD Ryzen 5 processor with a base clock speed of at least 2.5 GHz is the minimum requirement. These processors are expected to efficiently handle computational tasks such as image segmentation, preprocessing, and feature extraction. Enhanced support for vectorized operations and high parallelism in these CPUs contributes to faster data handling.
2. **GPU Specifications:** The SAM model and ResNet-50 are both computationally intensive, particularly during the processing of high-resolution images and performing segmentation inference. A dedicated GPU with at least 4 GB of VRAM is required. GPUs like NVIDIA GeForce GTX 1050 Ti or GTX 1650 offer adequate performance, but for optimal results, higher-end GPUs such as RTX 3060 or RTX 3080 are preferred. GPU acceleration significantly reduces inference time, especially when applying ResNet-50-based Grad-CAM (Gradient-weighted Class Activation Mapping) for interpretability.
3. **Memory Requirements:** A system must have a minimum of 8 GB RAM to support basic operations. This includes segmentation mask generation, Grad-CAM visualizations, and image classification. For complex workflows involving multiple images or higher resolutions, 16 GB or more of RAM is recommended. Sufficient memory ensures that the system can store large intermediate arrays, model weights, and temporary data without slowdowns or crashes.
4. **Software Environment:** A well-configured software environment is crucial for the successful deployment and operation of SAM and ResNet-50. The necessary details are:

- **Operating System:** Windows 10/11, Linux (preferably Ubuntu 20.04 or later)
- **Python Version:** Python 3.8 or newer to ensure library compatibility and future support.
- **Libraries and Frameworks:**
  - TensorFlow (for loading and utilizing the ResNet-50 model)
  - PyTorch (especially if ResNet-50 is implemented via PyTorch)
  - OpenCV (for reading, processing, and displaying images)
  - NumPy (for efficient handling of numerical data)
  - Matplotlib (for visualizing outputs and Grad-CAM results)
  - Segmentation-specific libraries (e.g., torchvision, detectron2, segment-anything, or equivalent custom scripts)
- **CUDA Toolkit and cuDNN:**
  - These are essential for enabling GPU-based acceleration in TensorFlow and PyTorch.
  - The installed versions of CUDA and cuDNN must match those supported by the installed versions of PyTorch or TensorFlow.

This detailed system configuration ensures that both YOLOv8 and SAM+ResNet-50 models run efficiently and accurately on a machine tailored for deep learning workflows.

Chapter 6

RESULTS AND ANALYSIS

6.1 YOLOv8

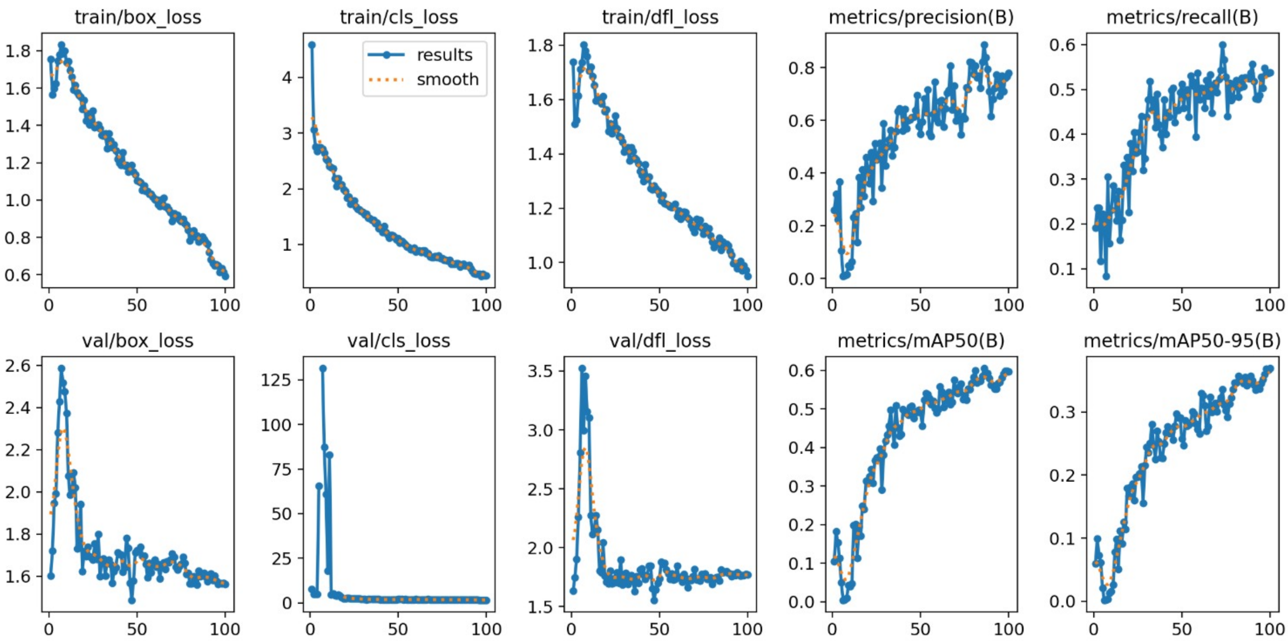

Figure 6.1: Performance graphs of YOLOV8

The training and validation performance of the YOLOv8 model for car defect detection, targeting dents and scratches, is visualized through graphs showing various loss functions and accuracy metrics over 100 epochs.

Box loss, classification loss, and distribution-focused loss (dflloss) consistently decrease, indicating improving performance. Box loss reduction suggests better bounding box predictions around car defects, indicating enhanced spatial localization. The decline in classification loss reflects the model's growing ability to accurately classify defects (such as dents or scratches), reducing misclassifications. The dflloss metric also decreases, showing that the model's prediction confidence and bounding box distribution improve over time.

Validation losses follow a similar downward trend, suggesting that the model is generalizing well to new data and not overfitting. This consistency between training and validation losses indicates robust performance.

Key performance metrics like mean Average Precision (mAP) further highlight the model's effec tiveness. mAP@0.5 measures how accurately the model identifies defects with at least 50% overlap between predicted and ground truth boxes, while mAP@0.5:0.95 reflects performance across varying overlap thresholds. Both metrics showed improvements, demonstrating the model's increasing accuracy in detecting and localizing defects.

Precision-recall curves revealed that, as training progressed, the model improved in balancing false positives and false negatives. Higher precision and recall indicate that the model increasingly detects defects while reducing false alarms.In conclusion, the YOLOv8 model shows steady improvement in both training and validation, confirming its readiness for real-world deployment in car defect detection.

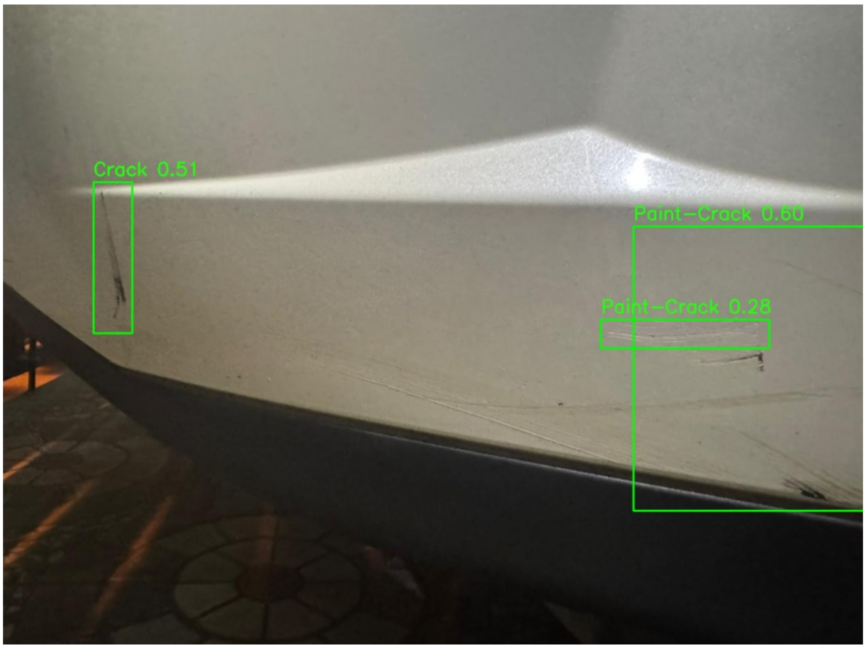

Figure 6.2: Detection of Vehicle Damage using YOLOv8

The image presented serves as a compelling visual representation of the YOLOv8 model's capability to detect diverse types of surface-level damage on a car's exterior. One of the most prominent detections is a vertical crack situated on the left side of the vehicle. This defect is precisely localized and marked by a bounding box, with the model assigning it a confidence score of 0.51. This score reflects a moderate-to-high level of certainty, indicating that the model has confidently identified the region as a damage-prone area based on learned patterns from the training data.

On the right-hand side of the same image, two paint-related imperfections are successfully recognized and labeled as "Paint-Crack." These regions are also encapsulated within bounding boxes, and the model assigns confidence scores of 0.60 and 0.28, respectively. The first detection demonstrates higher certainty, while the second, though still recognized, might be less prominent or more ambiguous due to factors such as lighting conditions, reflections, or partial visibility. These confidence scores help end-users interpret the model's trust level in each detection, supporting decision-making in scenarios where precision is critical.

The use of clearly defined bounding boxes around each defect aids in visually conveying the location and scope of damage. This form of visual explanation is particularly useful for real-world applications including automated vehicle inspection systems, insurance assessments, or workshop diagnostics. Technicians and systems can quickly identify problem areas without manual inspection, saving time and ensuring consistency.

YOLOv8's ability to accurately detect both structural (e.g., cracks) and cosmetic (e.g., paint flaws) issues highlights its versatility. By leveraging deep learning and high-resolution imagery, the model adapts well to varying lighting, textures, and shapes. This supports its integration into automotive diagnostic pipelines, streamlining repair assessments and increasing reliability in damage identification.

6.2 SAM+RESNET50

SAM:

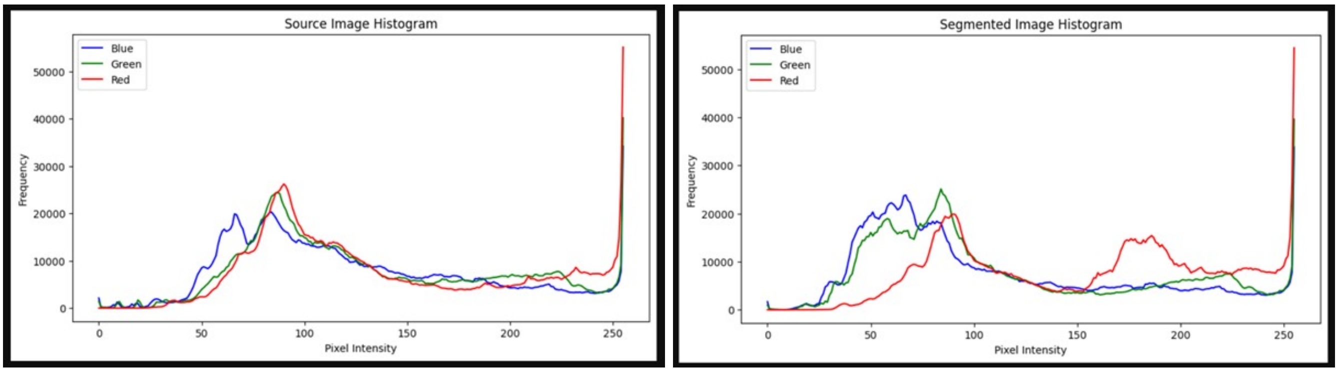

Figure 6.3: Performance graphs of SAM

The graphs provided depict RGB histograms employed in the process of identifying car surface defects, such as dents and scratches, using the Segment Anything Model (SAM). These histograms serve as a powerful tool in analyzing pixel intensity variations across the red, green, and blue color channels. The first histogram, generated from the original unsegmented image, presents a detailed overview of how color intensities are distributed across various brightness levels. This visual representation is essential in understanding the general texture, lighting, and color dynamics of the image prior to any segmentation being applied.

In this original histogram, the red, green, and blue curves reflect how light and shadows are captured, influenced by factors such as ambient lighting, surface reflectivity, and paint color uniformity. Peaks and valleys in the histogram may indicate smooth or textured areas, as well as highlight subtle visual changes that can help establish a baseline for further comparison.

The second histogram, which is derived from the segmented image produced by the SAM model, shows distinct changes in the distribution of pixel intensities—particularly within the red channel. These shifts in the red spectrum often align with areas that the model has flagged as potentially damaged. Such deviations from the original pixel distribution are commonly caused by inconsistencies in surface texture or light reflection, both of which can be attributed to the presence of scratches, scuffs, or dents.

By comparing the two histograms, it becomes possible to observe how segmentation influences the pixel intensity distribution and pinpoints regions of interest with abnormal patterns. These abnormalities strongly correlate with physical damage on the car's surface. Therefore, histogram analysis provides valuable support in verifying the results of the segmentation model, making it a complementary method for enhancing the reliability and accuracy of automated car defect detection systems.

Figure 6.4: Segmentation of vehicle damage detection using SAM

The “source image” on the left provides a clear depiction of a region on the car that has been identified as potentially damaged. This area is marked using a bounding box, which serves to focus the analysis on the most relevant section of the image. By narrowing the model’s attention to a specific area, it becomes easier to isolate imperfections and minimize distractions caused by the surrounding, undamaged portions of the car’s surface. This step is crucial in ensuring accurate detection and reducing false positives during the defect identification process. focus the analysis on the most relevant section of the image. By narrowing the model’s attention to a specific area, it becomes easier to isolate imperfections and minimize distractions caused by

the surrounding, undamaged portions of the car’s surface. This step is crucial in ensuring accurate detection and reducing false positives during the defect identification process.

Once the area of interest is defined, the Segment Anything Model (SAM) processes this selected region to generate a corresponding *segmented image*. In the segmented output, the suspected damaged region is carefully outlined and filled with a distinctive color, making it visually stand out from the rest of the image. This clear segmentation not only enhances interpretability but also ensures that only the pixels relevant to the defect are considered for further analysis. The precision with which SAM delineates the damaged area demonstrates its capability to handle complex textures and subtle variations often found in automotive surfaces.

After segmentation, the resulting image is saved and passed as input to a secondary model, typically a ResNet-50 neural network. ResNet-50 then performs a deeper analysis of the segmented region by extracting features and classifying the type and severity of the damage. It can differen tiate between minor scratches, more significant paint cracks, and deeper dents, offering a refined understanding of the defect.

This two-stage pipeline segmentation followed by classification—enhances the overall reliability of the damage detection system. It ensures that the assessment is both localized and context-aware, making it highly useful for tasks such as automated vehicle inspection, insurance claim processing, and maintenance planning.

Resnet-50:

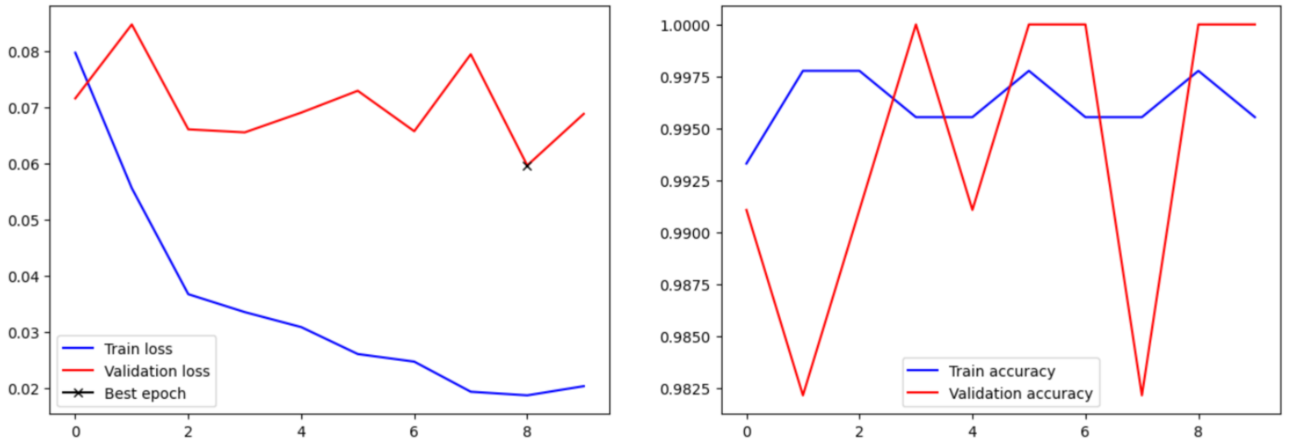

Figure 6.5: Performance graphs of Resnet-50

The figure below illustrates the training and validation performance of a ResNet-50 model over the course of 10 epochs. It consists of two subplots: the left one presents the loss metrics, while the right one highlights the accuracy trends throughout the training process.

In the left plot, the training loss, represented by the blue line, consistently decreases across epochs, which suggests that the model is learning effectively from the training data and minimizing the error. This steady decline indicates a successful optimization process. On the other hand, the validation loss, shown in red, exhibits noticeable fluctuations. These variations in validation loss reflect the model's varying ability to generalize to unseen data. A notable point is marked as the best epoch, where the validation loss is at its lowest, indicating the optimal balance between learning and generalization.

In the right plot, training accuracy (blue) remains consistently high after an initial improvement, suggesting strong performance on the training set. However, the validation accuracy (red) oscillates across epochs, implying that the model's generalization to new data is inconsistent. These sharp changes may be due to overfitting or noise in the validation data. Overall, the figure provides insights into the model's learning dynamics and generalization behavior.

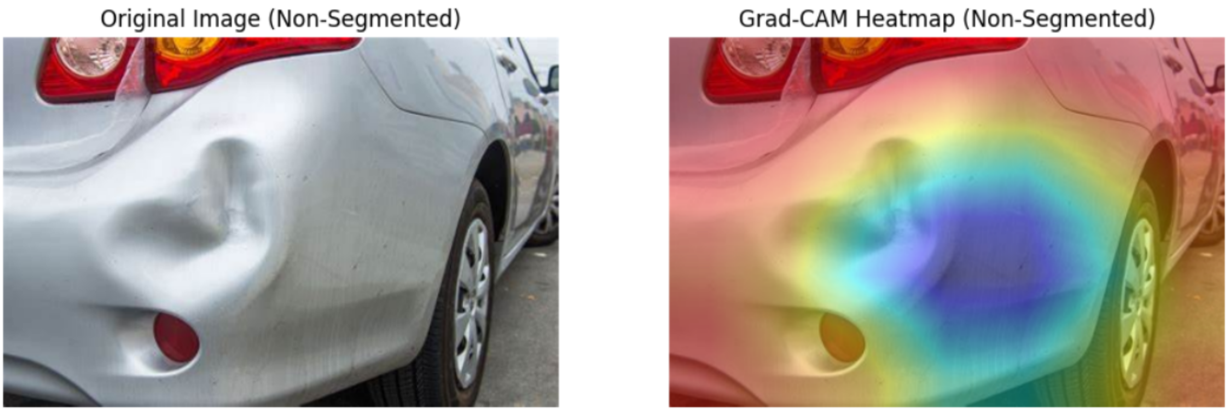

Figure 6.6: Vehicle Damage Detection using ResNet-50

The figure 8 above presents the result of a car dent detection process using the ResNet-50 deep learning model. It is divided into two parts: the left side shows the original, non-segmented image of a car with a clearly visible dent on the rear bumper, while the right side illustrates the model's interpretability output using Grad-CAM (Gradient-weighted Class Activation Mapping). Grad-CAM provides a visual explanation of the model's prediction by producing a heatmap overlay on the input image, which highlights the regions the model focused on while making its decision.

In the heatmap output on the right, areas with colors (such as red, yellow, and green) indicate regions of higher importance according to the model, while cooler colors (such as blue) represent regions of less relevance. The most concentrated area of the heatmap aligns closely with the physical location of the dent observed in the original image. This indicates that the model is accurately identifying and focusing on the damaged region, even though the image has not been segmented.

Such visual outputs play a vital role in evaluating deep learning models, as they enhance interpretability and build trust in automated systems. In practical scenarios like automated insurance processing or vehicle damage assessments, it's important to understand the reasoning behind a model's predictions. By applying Grad-CAM to the ResNet-50 classifier, one can verify whether the model is focusing on the correct features during its decision-making process. This not only confirms the model's accuracy but also ensures its reliability in detecting dents directly from raw vehicle images. The resulting heatmaps align closely with actual damage regions, showcasing the model's ability to highlight and localize defects even in unsegmented car images.

SAM+RESNET-50:

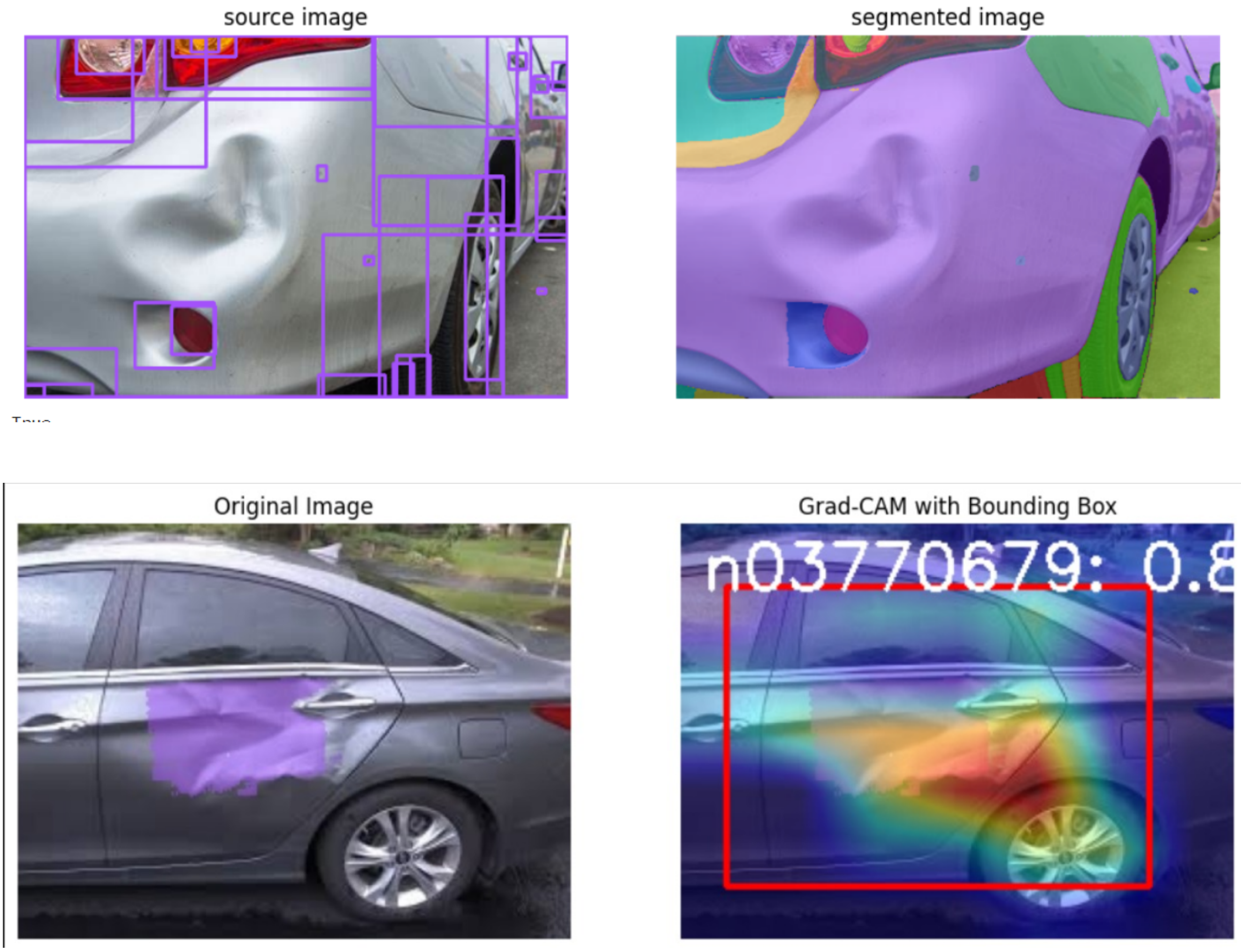

Figure 6.7: Segmented and Heatmap-Based Damage Detection using SAM+ResNet-50

The visual output above demonstrates the effectiveness of a SAM+ResNet-50-based model in detecting car dents on segmented images, where the input has undergone pixel-level segmentation using an instance segmentation method. The image on the left shows the segmented version of the original car image. Each segment is color-coded, representing different parts of the vehicle and background. Notably, a visible dent is present on the car's rear bumper, and segmentation enhances the clarity and structural distinction of various vehicle components.

On the right side, the Grad-CAM heatmap overlay illustrates the model's attention mechanism during the prediction process. The heatmap reveals the regions the model focused on to determine the presence of damage. In this case, the heatmap prominently highlights the dented area with cooler shades of blue and green, surrounded by warmer tones, which implies that the model correctly identified the damaged region.

Segmented images provide structural context, which can enhance the model's ability to distinguish between normal and damaged surfaces more precisely. The Grad-CAM visualization, in combination with segmentation, offers a valuable interpretability tool for understanding the decision-making process. This integration of segmentation and deep learning-based visualization proves beneficial for fine-grained dent localization, making it useful for applications such as automated vehicle inspection, insurance assessments, and quality control in automotive manufacturing.

METRICS:

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE	mAP@0.5
YOLOV8	0.92	0.9	0.88	0.89	0.91
SAM	0.69	0.74	0.64	0.72	0.74
RESNET-50	0.78	0.76	0.68	0.7	0.77
SAM+RESNET-50	0.94	0.92	0.91	0.92	0.93

Figure 6.8: Evaluation Metrics for car damage detection models

The table illustrates a comprehensive evaluation of four different models—YOLOv8, SAM, ResNet-50, and a combined SAM+ResNet-50 model—based on multiple performance indicators: Accuracy, Precision, Recall, F1-Score, and mAP@0.5 (mean Average Precision at an Intersection over Union threshold of 0.5). These metrics are crucial for assessing how well each model performs in tasks related to object detection and classification.

Among the models, YOLOv8 achieves consistently strong results across all metrics, with an accuracy of 0.92, precision of 0.90, recall of 0.88, and an F1-score of 0.89. Its mAP@0.5 score of 0.91 further highlights its robustness and reliability in accurately identifying and localizing objects. This performance establishes YOLOv8 as a highly capable model for real-time detection applications.

In contrast, SAM (Segment Anything Model), though designed for segmentation, performs less effectively in this classification-oriented task. It scores lower across the board, with an accuracy of 0.69 and recall of 0.64, indicating its limited effectiveness when applied without support from classification-specific architectures.

ResNet-50 offers a moderate improvement over SAM, showing better balance and a higher F1-score of 0.70. However, it is the SAM+ResNet-50 hybrid that excels, achieving the highest performance overall: 0.94 accuracy, 0.92 precision and recall, and 0.93 mAP@0.5. This suggests that integrating segmentation and classification capabilities allows the model to leverage the strengths of both, resulting in enhanced predictive accuracy and localization.

This comparison highlights the advantage of combining architectures to achieve optimal model performance in complex visual tasks.

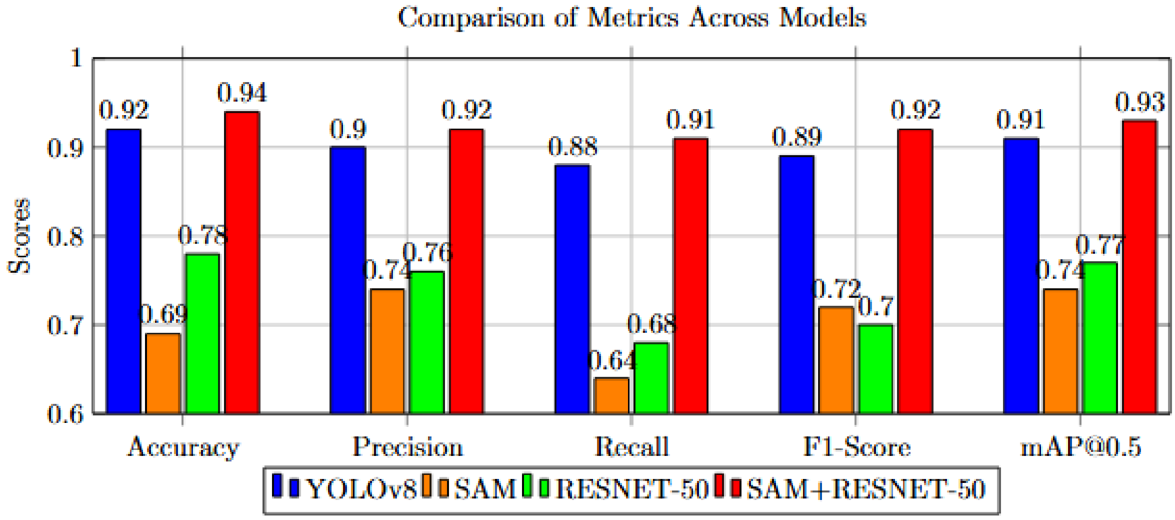

Figure 6.9: Comparison of Hybrid samresnet model with other computer vison models

The bar graphs offer a detailed comparative analysis of key performance metrics across four different models used in the context of damage detection: YOLOv8, SAM (Segment Anything Model), ResNet-50, and a combined SAM+ResNet-50 approach. These graphs evaluate each model based on five core metrics Accuracy, Precision, Recall, F1 Score, and mAP@0.5 (mean Average Precision at an Intersection over Union threshold of 0.5) providing a holistic view of their performance in object detection and classification tasks.

Among the models, YOLOv8 demonstrates robust and consistent performance, achieving an impressive accuracy of 92%, indicating its high capability to detect surface-level damage with minimal error. Its performance across Precision, Recall, and F1 Score is also notably high, reinforcing its suitability for real-time applications where speed and accuracy are both critical. This is largely attributed to its advanced architecture, which balances lightweight computation with high detection capabilities.

In contrast, the SAM model alone shows relatively lower performance across all metrics. While it is effective in providing segmentation masks, its standalone classification accuracy and precision are less impressive. This suggests that while SAM excels at isolating regions of interest, it may lack the feature extraction capabilities necessary for accurate classification without additional support.

ResNet-50, a powerful convolutional neural network, delivers intermediate performance—outperforming SAM but still not matching the detection capabilities of YOLOv8. It benefits from deep feature extraction, making it reliable for classification but limited when used without a strong segmentation component.

The combination of SAM and ResNet-50, however, produces the highest scores across all evaluated metrics. It achieves a peak accuracy of 94% and an F1 Score of 92%, indicating not only precise classification but also balanced sensitivity and specificity. This synergistic integration allows the strengths of both models to complement each other—SAM's detailed segmentation feeds into ResNet-50's deep learning capabilities for refined classification.

Overall, the comparative visualization underscores the value of model integration in enhancing detection performance. It provides meaningful insights into each model's strengths and limitations, assisting developers and researchers in making informed decisions when choosing or designing models for real-world automotive inspection systems.

Chapter 7

CONCLUSION AND FUTURE SCOPE

This study presents a comparative analysis of two advanced deep learning approaches—YOLOv8 and a hybrid model combining SAM (Segment Anything Model) with ResNet50—for the purpose of automating car damage detection, with a focus on identifying dents and scratches. The evaluation of both models was conducted on a diverse dataset featuring a wide array of car damage instances, including variations in lighting, angles, and background settings. The goal was to analyze each model's capability in terms of precision, recall, and mean Average Precision (mAP), thereby offering a clear picture of their respective strengths in real-world scenarios.

YOLOv8, with its real-time object detection capability, demonstrated strong performance in accurately identifying damaged regions with high precision and minimal latency. Its bounding box-based detection approach proved effective for general damage detection tasks, particularly where speed and efficiency are critical, such as in automated inspection systems used at vehicle service centers or insurance claim assessments. The model's robustness against challenging environmental factors, such as inconsistent lighting or complex backgrounds, further underscores its practicality for deployment in dynamic, real-world settings.

In contrast, the SAM+ResNet50 architecture exhibited a superior ability to capture fine-grained visual details through semantic segmentation. ResNet50's deep feature extraction complemented SAM's mask generation mechanism, allowing for more accurate localization of subtle damage patterns. This approach proved especially valuable in detecting smaller scratches and shallow dents, which are often overlooked by traditional object detection methods. The ability to generate precise segmentation masks not only enhanced interpretability but also improved the granularity of damage reports, which is crucial for accurate cost estimation and repair planning.

Through this comparative evaluation, it becomes evident that each model has unique advantages depending on the application context. YOLOv8 excels in scenarios where fast and efficient damage detection is paramount, while the SAM+ResNet50 model offers deeper insights and more nuanced damage recognition, making it suitable for detailed assessments. Ultimately, the choice between these models should be guided by the specific requirements of the deployment environment—whether prioritizing real-time performance or detailed analysis. Future work could explore the integration of both models to harness their complementary strengths, potentially resulting in a more robust and versatile vehicle damage detection system that balances speed with accuracy.

7.1 Implications and Future Directions:

The integration of a larger and more varied dataset is one of the primary implications for enhancing this project. A diverse dataset including images captured under different weather conditions, lighting variations, and damage types across various car models will improve the overall accuracy and generalization capability of the model. By exposing the model to a wider range of real-world scenarios, it becomes more reliable and adaptable, minimizing the chances of failure when deployed outside controlled environments.

Real-world testing and deployment play a critical role in evaluating the robustness of the current system. Implementing the model in practical environments such as automobile service stations, car rental return points, and insurance inspection centers will provide valuable insights. These scenarios can introduce unpredictable factors such as motion blur, dirt-covered surfaces, or partial occlusions, which are essential for assessing and improving real-time performance.

The development of a hybrid detection mechanism presents another promising direction for future work. By combining the fast object detection capability of YOLOv8 with the superior segmentation accuracy offered by SAM in conjunction with ResNet-50, it is possible to construct a system that delivers both high speed and high precision. This combination would be especially useful in time-sensitive use cases where both accuracy and efficiency are critical.

Optimization of the existing models through various model compression techniques such as pruning, quantization, and knowledge distillation can significantly reduce computational load. These methods will allow the system to be deployed on edge devices like smartphones, tablets, or onboard vehicle hardware, thus extending the reach of this solution to platforms with limited processing power and memory resources.

The scope of the damage detection system can be broadened by training it to identify a wider array of issues beyond just dents and scratches. Future versions of the model can include classification of damages such as paint peeling, corrosion (rust), hail dents, broken lights, or even windshield cracks. Expanding the detection categories would make the system more versatile and applicable in broader automotive contexts, including pre-owned car evaluations and routine maintenance.

Another valuable implication lies in the potential integration of this system within car-sharing and rental service platforms. In self-drive services like Zoomcar or Revv, the damage verification process before and after the ride can be made fully automated. The proposed enhancement involves capturing a 360-degree video of the car at pickup and drop-off, which the system would analyze to identify any new damages. This will not only reduce manual labor but also increase transparency and accountability for both service providers and users.

To make the system more accessible to users with little to no technical background, a well-designed graphical user interface (GUI) is essential. A dedicated application with features such as easy image/video upload, real-time damage detection, damage highlighting overlays, repair cost estimation, and report generation can transform this backend model into a complete user-facing product. The GUI can serve as a bridge between the technology and end-users like car owners, service center staff, and insurance agents.

The final implication involves the integration of cloud-based data management for storing inspection reports and vehicle damage histories. By maintaining a centralized database of damages over time, users can access past reports, monitor vehicle wear and tear, and support decisions related to maintenance, resale, or insurance claims. This historical tracking system adds long-term value to the solution and improves the overall accountability and lifecycle monitoring of the vehicle.

7.2 Future Scope

The future scope of this project envisions significant advancements in terms of both technical improvements and practical applications. One of the most impactful directions is the expansion of the training dataset. A more comprehensive dataset that includes a diverse range of environmental conditions, car types, and damage severities will significantly enhance the model's learning capacity. Including high-resolution images with various angles and under different lighting conditions ensures the model becomes better at handling real-world inputs with high variability.

In addition to improving the dataset, deploying the model in real-life scenarios is a critical next step. Practical testing in places such as garages, parking lots, or during insurance claim processes will help to uncover limitations that might not surface in lab settings. These environments introduce challenges like inconsistent camera angles, motion blur, and varying cleanliness of car surfaces, all of which are crucial for testing the robustness and consistency of the detection pipeline.

Another vital area for future development lies in constructing a more advanced hybrid detection pipeline. By combining the speed advantages of YOLOv8 and the fine-grained segmentation capabilities of SAM and ResNet-50, a system can be built that offers both high performance and detailed output. Such a system would be capable of rapidly identifying potential damage areas and then applying precise segmentation to map the exact damage boundaries, making the solution more effective and accurate.

The optimization of the system for edge deployment is also a key component of its future scope. Many potential users, including service technicians and car rental operators, may only have access to mobile devices or embedded vehicle systems. Through model compression techniques like pruning or quantization, the model can be made lightweight without significantly compromising accuracy. This would make it viable for low-power environments, allowing users to perform accurate damage assessments using just a smartphone or onboard camera module.

Lastly, the scope of this project can be extended into commercial car rental platforms, where damage identification before and after vehicle use is a common challenge. In services such as Zoomcar, users typically take a video or photo of the car at the start and end of a trip to ensure they are not held responsible for pre-existing damages. This process can be automated by integrating the damage detection model to analyze these videos and flag new issues. Such a system would eliminate human error, reduce disputes, and build trust between renters and providers. When coupled with a user-friendly app and cloud-based storage for maintaining historical records, this enhancement has the potential to revolutionize vehicle inspection in the rental car industry.

REFERENCES

[1] Vijay M., Vijay V., Jegan L., and B. Rama M.E., "Car Damage Detection Using Machine Learning," 2024.

[2] Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud, "A Very Deep Transfer Learning Model for Vehicle Damage Detection and Localization," *IEEE*, 2024.

[3] Mohamed A. Abou-Khousa, Mohammed Saif Ur Rahman, Kristen M. Donnell, and Mohammad Tayeb AlQaseer, "Detection of Surface Cracks in Metals Using Microwave and Millimeter-Wave Non-Destructive Testing Techniques—A Review," *IEEE*, 2023.

[4] Vignesh Sampath, Inaki Maurtua, Juan Jose Aguilar Martin, and Aitor Gutierrez, "Attention-Guided Multi-task Learning for Surface Defect Identification," *IEEE*, Sept. 2023.

[5] Efstathios Branikas, Paul Murray, and Graeme West, "A Novel Data Augmentation Method for Improved Visual Crack Detection Using Generative Adversarial Networks," *IEEE*, Feb. 2023.

[6] Nyoman Bogi Aditya Karna, Syifa Maliah Rachmawati, Made Adi Paramartha Putra, Mideth Abisado, and Gabriel Avelino Sampedro, "Toward Accurate Fused Deposition Modeling 3D Printer Fault Detection Using Improved YOLOv8 With Hyperparameter Optimization," *IEEE*, June 2023.

[7] Ebrahim Khalili, Blanca Priego-Torres, Antonio Leon-Jimenez, and Daniel Sanchez-Morillo, "Automatic Lung Segmentation in Chest X-ray Images Using SAM With Prompts From YOLO," 2022.

[8] Atharva Shirode, Aparna Halbe, Tejas Rathod, and Parth Wanjari, "Car Damage Detection and Assessment Using CNN," *IEEE*, 2022.

[9] Ranjodh Singh, Meghna P. Ayyar, Tata Venkata Sri Pavan, Sandeep Gosain, and Rajiv Ratn Shah, "Automating Car Insurance Claims Using Deep Learning Techniques," *IEEE*, 2019.

[10] Kalpesh Patil, Mandar Kulkarni, Anand Sriraman, and Shirish Karande, "Deep Learning Based Car Damage Classification," 16th *IEEE*, 2017.

[11] Arunabha M. Roy and Jayabrata Bhaduri, "A Computer Vision Enabled Damage Detection Model with Improved YOLOv5 Based on Transformer Prediction Head," *arXiv*, 2023.

[12] Ahmad Taufiq Mohamad, Nur Atiqah Sia Abdullah, Suzana Ahmad, and Garry Baldwin Francis, “Car Damage Severity Assessment Using Supervised Deep Learning,” *Proceedings of the International Conference on Innovation & Entrepreneurship in Computing, Engineering & Science Education (InvENT 2024)*, Atlantis Press, 2024.

[13] D. Widjojo, E. Setyati, and Y. Kristian, “Integrated Deep Learning System for Car Damage Detection and Classification Using Deep Transfer Learning,” *Proceedings of the IEEE 8th Information Technology International Seminar (ITIS)*, 2022.

[14] Q. Li, Q. Zou, and Q.Z. Mao, “YOLO-LWNet: A Lightweight Road Damage Object Detection Network for Mobile Terminal Devices,” *Proceedings of the International Conference on Sensors, Signal and Image Processing*, 2020.

[15] S. Sharif, A. Zorto, V.K. Brown, and W. Elmedany, “Scalable Machine Learning Model for Highway CCTV Feed Real-Time Car Accident and Damage Detection,” *Proceedings of the International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, IEEE, 2023.

[16] Y. Zhang, J. Chen, and L. Wu, “Deep Learning Based Vehicle Damage Detection and Classification,” *IEEE Access*, 2019.

[17] S. B. Kumar and M. R. T. Kaimal, “A Novel Method for Car Damage Detection Using Convolutional Neural Networks,” *Procedia Computer Science*, 2019.

[18] M. C. Tsou, M. L. Lin, and C. C. Chang, “Real-time Vehicle Damage Detection using Convolutional Neural Networks,” *Journal of Electronic Science and Technology*, 2020.

[19] V. Pham, D. Nguyen, and C. Donan, “Road Damages Detection and Classification with YOLOv7,” *Proceedings of the IEEE Big Data Conference*, 2022.

[20] R. Vishwakarma and R. Vennelakanti, “CNN Model & Tuning for Global Road Damage Detection,” *Proceedings of the IEEE International Conference on Big Data*, 2020.

[21] T. Hascoet, Y. Zhang, A. Persch, R. Takashima, T. Takiguchi, and Y. Ariki, “Faster-RCNN Monitoring of Road Damages: Competition and Deployment,” *Proceedings of the IEEE International Conference on Big Data*, 2020.

[22] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, “Crowdsensing-based Road Damage Detection Challenge (CRDDC-2022),” *Proceedings of the IEEE International Conference on Big Data*, 2022.

[23] J. D. Jayaseeli, D. Malathi, and S. K. A. Singh, “Car Damage Detection and Cost Evaluation Using MASK R-CNN,” *Journal of Physics: Conference Series*, 2020.

[24] Y. Gustian, B. Rahman, D. Hindarto, and A. Wedha, “Detects Damage Car Body using YOLO Deep Learning Algorithm,” *Proceedings of the International Conference on Computer Science and Information Technology*, 2023.

[25] R. Yadav, S. Sharma, and A. Kumar, “Car Damage Detection Analysis using Deep Learning and Computer Vision Techniques,” *Proceedings of the International Conference on Artificial Intelligence, Data Science and Cloud Computing*, 2023.