CAR CONDITION DETECTION USING CNN ALGORITHM

MSc Research Project
MSc in Data Analytics

Keerthana Reddy Mettu
Student ID: X23275634

School of Computing
National College of Ireland

Supervisor: Jaswinder Singh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Keerthana Reddy Mettu
Student ID:	23275634
Programme:	MSc in Data Analytics **Year:** 2024 - 2025
Module:	MSc (Research) Practicum 2
Supervisor:	Jaswinder Singh
Submission Due Date:	15-09-2025
Project Title:	Car Condition Detection using CNN Algorithm
Word Count:	8004 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Keerthana Reddy Mettu

Date: 15-09-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

CAR CONDITION DETECTION USING CNN ALGORITHM

Keerthana Reddy Mettu

x23275634

Abstract

The paper explores the implementation of damage detection in automated vehicles represented by Convolutional Neural Networks (CNNs) and assesses three different architectures including, Custom CNN, ResNet18, and VGG16, used under standardized preprocessing conditions. The goal was to examine the effects of both the architecture depth and transfer learning on the accuracy and robustness of classification. A collection containing pictures of cars, labeled according to the type of damages, was normalised, resized, and augmented to promote generalization. The difference in performance was found to be significant as a result of experimental tests. The Custom CNN performed at 47% accuracy, meaning that it is not effective to deal with the subtle and complex patterns of damages. ResNet18 increased accuracy to 69%, and it was again good clear damage, but inconsistent borderline classes. VGG16 proved to be better than the two with 82% accuracy and showed to be balanced in F1-scores and to be highly adaptable when conditions change. These results point to the efficacy of deep, and pre-trained CNNs in visual damage evaluating duties. In future, segmentation methods, ensemble modeling, and bigger and more varied datasets will be incorporated into the work in order to increase precision and applicability in a practical scenario. Moreover, the explainable AI will lead to better interpretation and confidence in automations. This study will add to already existing practical, AI-based insurance claims, fleet and vehicle inspections, and vehicular maintenance supporting more significant efficiency with precise and faster assessment of the vehicle condition at a lower cost.

1 Introduction

1.1 Background/motivation

Automated vehicle inspection systems have become possible due to the progress in deep learning and computer vision in recent times. Nevertheless, the majority of current research addresses only general damage detection but does not consider the difficulties of the real-world variability (inconsistent lighting, complex backgrounds, partial occlusions, etc.). This study is the first of its kind to directly test CNN architectures with these practical limitations, with the goal of providing practical solutions that can be used in automotive inspection application areas. Image processing and spotting damages in cars are best handled by CNNs in the automotive industry.

1.2 Aim and objectives

Aim

The goal of this study focuses on improving vehicle damage detection by analysing CNN architectures and preprocessing methods along with segmentation models to enhance accuracy and robustness in real-life automated vehicle checks using Python programming, with an emphasis on ensuring consistent performance across varied environmental conditions and practical operational settings.

Objectives

- To study how varying image preprocessing approaches influence CNN performance in several testing environments.
- To analyse different CNN architectural formats to identify which one delivers optimal vehicle damage classification results.
- To explore damage detection through segmentation and its integration with CNN classification systems.
- To design and implement a well-defined evaluation framework that measures not only classification accuracy but also computational efficiency, scalability, and adaptability to diverse real-world use cases such as insurance claims processing and fleet inspections.

1.3 Research Questions

1. How do different CNN architectures perform in vehicle damage detection when the same preprocessing technique is applied under varying lighting conditions?
2. Which CNN architecture yields the highest classification accuracy (or performance metric) for vehicle damage detection using a consistent preprocessing pipeline?

1.4 Significance of the study

This project aims to resolve the urgent need for tools to check a vehicle's condition swiftly and fluently. In contrast to generic solutions, the proposed solution is application-specific and suitable to be implemented into the processes of the automotive industry, where a quick, precise, and reliable inspection is of importance. The research helps to find out how to create strong damage detection systems by focusing on weaknesses brought upon by poor lighting systems, noise distractions, and incomplete obstruction. The results can be directly applied in enhancing efficiency in insurances evaluation, in the time taken in manual inspection, and making decision-making in fleets management processes. Challenges appear in practical use because inferior lighting, unwanted noises, and blocked parts of an image can affect the quality of the process. Therefore, these challenges may reduce how dependable CNN models are.

1.5 Approach or gap in literature

It is proposed in the study to build a unique CNN model, together with using advanced preprocessing methods to address these issues. It tries to fill existing knowledge gaps by assessing and enhancing the stability of models in many different imaging situations. This research helps in the field of insurance and also gives practical results for fleet management and automotive inspection.

2 Related Work

This chapter critically analyses the current body of literature of car condition detection using CNNs, placing the current study into the general context of deep learning-based vehicle vision systems. The overview of the key advancements, strengths, and weaknesses of the recent approaches and the discussion of the main research gaps that inspire this study are provided.

2.1 Deep Learning for Vehicle Damage and Condition Detection

2.1.1 Real-Time Vehicle Damage Detection

Kulambayev et al. (2024) proposed a system of detection of road damages in real-time based on image analysis using CNN. The research aimed at identifying the structural failures like cracks and potholes with the help of deep learning pre-processing pipeline specific to roads. The fact that the system was capable of real-time applications also served as a great contribution, especially when it comes to applying in dynamically moving vehicles like autonomous driving, or road maintenance monitoring. The relevance of CNNs in real-life has been confirmed by the achieved high accuracy at detecting surface-level anomalies. Nevertheless, the size, variety, and type of damage that they captured limit the previously mentioned results. It was also the case that the majority of the data was collected on a few types of damages (potholes and cracks) and in general did not give the more granulated and type-specific information on damages to vehicles which could be the scratches, dents and glass which breaks (Alshehri *et al*. 2025). Moreover, this experiment was not properly conducted in a diverse atmosphere of light and multidimensional urban background, which impedes its applicability in such locations as the parking or other modes of inspection depots where these conditions are ordinary. Their preprocessing approach was equally not very developed, so there was a little idea about how it improves or hurts the performance of CNN.

2.1.2 Damage Detection with Cost Estimation

Nackathaya et al. (2024) extended the applicability of CNNs further, since they deployed a combined damage classification and repair cost estimation system. The kind of research is of high significance to the real world industries like automotive insurance, fleet management as well as service logistics. They had a dual-approach system the CNNs were to determine damage and the models based on regression could be able to determine the costs of repair which demonstrated a deep synergy between deep learning and the support of business decisions. This paper has shown that CNNs are well able to segment and identify types of damages including minor dents, major scratches, and lamps that have been broken with a relatively high precision. Nonetheless, the model performed poorly on images taken in non-homogenized sources of light, when there was a slight masking (through the shadows or any other object), or when there was an oblique view of the camera. These are scenarios that are common in actual inspection environments meaning that although the model was quite adequate in the controlled environments; its effectiveness in naturalistic contexts was not established (Fahad *et al*. 2025). Furthermore, there was no deeper investigation of any preprocessing methods, including such methods as normalization, contrast enhancement, and noise reduction, which are confirmed to have substantial impacts on the accuracy and robustness of CNN models.

2.1.3 Crash Severity Prediction Using CNNs

Even though Rahim and Hassan (2021) did not directly address the issue of vehicle condition detection, they made a contribution to the field as they implemented the framework based on deep learning to predict the severity of traffic crashes. They contribute to the current approaches with their efforts, because they highlight deep learning in the analysis of high-risk scenarios on vehicles, which requires interpretation of complex images and description of the context. They employed both image clues and additional signs (such as weather conditions and road conditions) to anticipate the type of crash outcome in terms of minor injuries, major injuries, and fatal ones. Another important methodological advantage consists of the use of processed data in the form of images since it contributed to better model accuracy and the elimination of noise-related mistakes (Guo *et* *al*. 2025). The pipeline involved such techniques as resizing, denoising, and normalization, which underline the importance of the preprocessing to achieve better CNN performance.

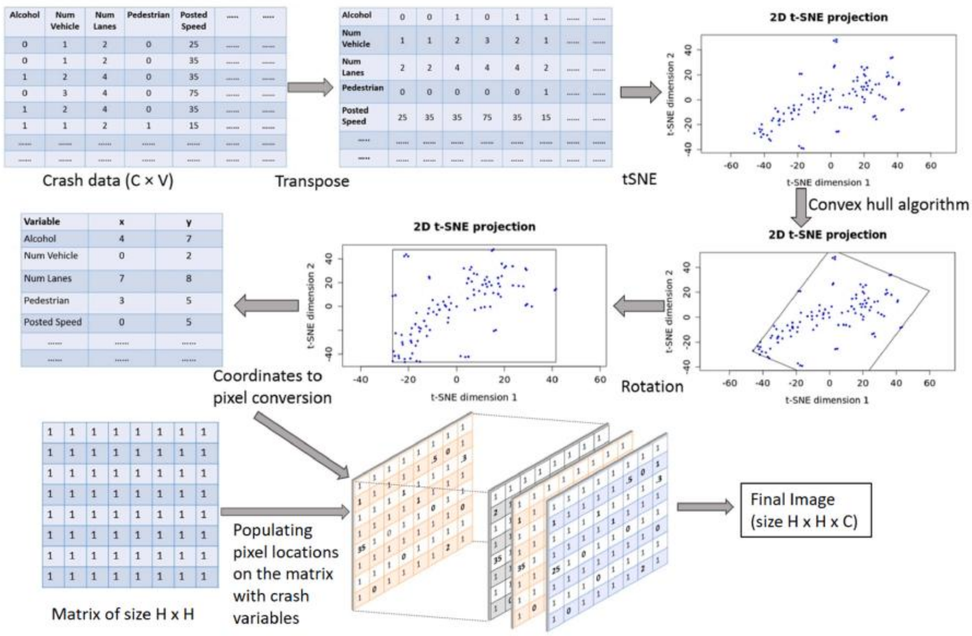

Figure 2.1: Workflow for converting numerical data into images

(Source: Rahim and Hassan, 2021)

The study itself however was mostly focused on crash severity and thereby does not go up to the level of damage to certain parts of a vehicle, hence does not apply to stating the condition of those parts. Moreover, there were no segmentation-based detection solutions, which could have used the localized features like damage extent on bumpers, doors, or windows. However, the fact that Rahim and Hassan incorporated factors of environmental and contextual data into the study proves to be a good learning lesson: deep learning systems should be built with due knowledge of the data variability availability including factors that are beyond vehicle body that could influence the accuracy of detection. The strengths of the study in preprocessing and feature integration lead to the fact that more comprehensive frameworks are needed in the detection of car conditions (Hu *et al*. 2025).

2.1.4 Vision-Based Systems for Autonomous Vehicles

Pavel et al. (2022) provided a detailed systematic literature review of vision-based self-driving vehicle systems and its use with deep learning. The paper is crucial when it comes to the topic of the functioning of CNNs in automobile contexts especially when it comes to the recognition of stimuli and its cause. Peer critical analysis of preprocessing data pipelines, including image normalization, augmentation, and noise filtering upon which model stability and accuracy rests forms one of the major contributions of the present study. Pavel et al. (2022), emphasized that it is necessary to take into account the diversity of datasets and train models on a variety of environmental conditions face-plating in cities, country areas, and highways. Nevertheless, they noted that most of the studies are still based on ideal condition categories; hence, are poorly applied when it comes to real-time application. In addition, they also considered that the segmentation models are not frequently incorporated with the classification CNNs in the domain of damage evaluation. That is why this observation is closely approached in our work that aims to fill this gap through combining segmentation methods of U-Net or Mask R-CNN with CNN-based classifier thus contributing to sensitizing damage localization (Wang, 2024). It is identified as one of the weaknesses of the review by Pavel et al., that it does not include empirical comparisons of performance of various CNN architectures (e.g., VGG, ResNet, Inception) across vehicle-specific tasks. This constrains the applied lessons that may be used by researchers in choosing an ideal model in dealing with certain damage detection work (Hasan and Dey, 2024). However, their work reaches the conclusion that more integrated end-to-end frameworks, which integrate several deep learning paradigms in different combinations to enhance performance in diverse vehicle inspection environments, are necessary.

2.1.5 Challenges in Deep Learning-Based Vehicle Recognition

Boukerche and Ma (2021) even widened the scope of the discussion by comparing the difficulties of recognizing autonomous vehicles by means of vision systems. In their review, they focused on how the variability in the environment, including lighting changes, weather conditions, occlusions, and others, can adversely affect the execution of CNN-based recognition systems. They are immediately applicable in vehicle damage detection systems where in order to perform effectively, the systems need not be limited by the conditions of the scene or the background. Their critical reviews showed that CNNs were vulnerable and that most of the models did not perform well due to the inability to generalize domain-wise. It was also explained by the unavailability of datasets and, also, effective preprocessing and domain adaptation approaches.

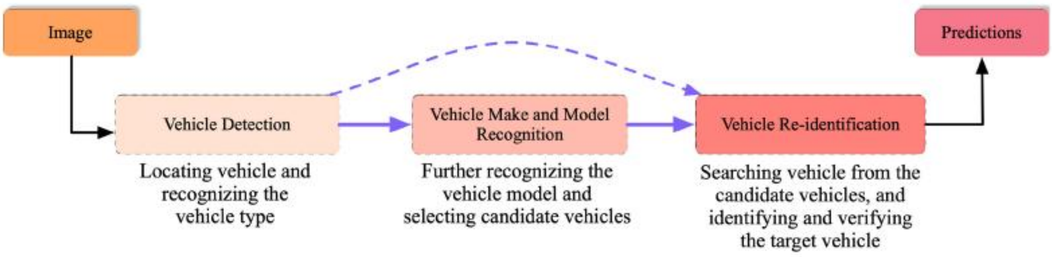

Figure 2.2: Process diagram illustrating numeric-to-image transformation

(Source: Boukerche and Ma, 2021)

A CNN trained on daytime sunny conditions might not perform well in the dark or in the rain. This kind of degradation of performance cannot be tolerated in real-life situations, like insurance rating of assets or quality-checking backgrounds done after accidents. Moreover, Boukerche and Ma (2021), emphasized the necessity of modular architectures that would support adaptive learning highlighting the importance of abandoning monolithic CNNs in favor of hybrid architectures that can group classification, detection, and segmentation in a single system. Their recommendations echo the framework of the present work which is going to develop the unified CNN model with the support of preprocessing pipelines and optional segmentation layers to evaluate the specific damage in a deep way. In their analysis, it came out that the majority of them did not carry out comparative benchmarking across various CNN architectures on a common preprocessing setting. This is not standardized to restrict reproducibility and verification of the model in various research. This observation gives us an impetus in pursuing our research agenda of controlled comparisons of CNN architectures (e.g., ResNet, VGG) under preTEXTURE: that the preprocessing techniques should be uniform (Liu *et al*. 2025).

2.2 Summary and Research Gap

According to recent research, CNN-based systems have increasingly been used in vehicle damage detection and produce promising results related to the localisation of road damage (Kulambayev et al., 2024), crash severity prediction (Rahim and Hassan, 2021) and autonomous vehicle vision (Boukerche and Ma, 2021). Nevertheless, there are still a number of gaps that are consistent. Most importantly, few attempts have systematically tested preprocessing methods commonly used to images, including normalization, contrast enhancement, and noise removal, though their effects on CNN stability and domain generalisation are known. Secondly, there is very limited integration between these two processes segmentation and classification, despite possible benefits to localisation and interpretability of instrumenting damage via methods such as U-Net. Third, the vast majority of works do not have standardised benchmarking, and it is hard to compare CNN-based solutions according to a similar preprocessing pipeline. Further the resistance to objections in real life (e.g., variable lighting, obstruction and background clutter) is commonly presumed as opposed to properly evaluated and deployment aspects are seldom ever tackled. This paper aims to fill these gaps by introducing a unified CNN architecture that joins together the processes of preprocessing, segmentation, and classification, and by making controlled comparisons of various models in diverse environmental settings to create a more reliable and deployable solution to the task of vehicle damage detection.

3 Research Methodology

3.1 Research Procedure and Experimental Design

The study uses an experimental research design and relies on the method of comparison to prove the efficiency of different CNN structures and the use of these models in preprocessing when detecting vehicle damage automatically. This process requires three major stages, which include preprocessing of the images, the implementation of the model, and verification of the results. All phases are systematically designed to follow a systematic observation of the effectiveness of each component on the performance of classification. The dataset of the images of vehicles with various levels and quality of damage is processed by utilising various methods of image enhancement (Tahir et al. 2024). These are the grey scale conversion, resizing, noise reduction, contrast normalisation and augmentation. The purpose is to mimic environmental variability encountered in reality, i.e., the low-light condition, noise in the image, or image occlusion.

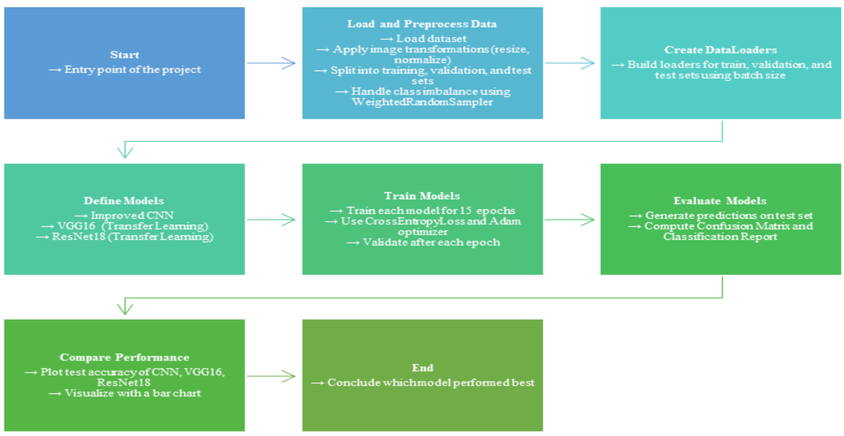

Figure 3.1: Deep Learning Pipeline for Image Classification Using Transfer Learning

(Source: Self-created)

The pre-processed datasets are used to implement and train multiple CNN architectures, including standard models, such as VGG16, ResNet18 (updated), and an improved custom CNN that incorporates Adaptive Average Pooling and dropout for regularisation, as well as transfer learning techniques. The models are tested on stable and variable lighting conditions to test robustness. In order to address class imbalance in the training data, a WeightedRandomSampler is implemented, ensuring equitable learning across categories (Ghosh, 2021). The same tests are carried out under the same conditions so that a comparison can be made unbiasedly. This is a very strict experimental plan, as it enables the study to establish the most efficient preprocessing strategies and CNN architectures that can be deployed in real-life vehicle inspection projects.

3.2 Data Collection and Processing Techniques

In order to achieve better results, the dataset provided in this paper features labelled, annotated images of vehicles and types of damage. The types of data used are forms of open-access repositories, publicly accessible vehicle damage sets, and images taken and captured with live time and different lighting and environmental conditions to increase the diversity of data sets (Ghosh, 2021). The preprocessing pipeline is extensive and is individually optimised to advance the overall CNN model's performance.

Pre-processing Steps

In this project, a systematic set of pre-processing procedures was used to prepare the data to train and test the model. These measures facilitated consistency amidst the inputs, mitigated class imbalance and also made the dataset optimally fit the convolutional neural network (CNN) models.

Normalization and image resizing

To ensure that the requirements of the model did not change across the various input images, each of them was converted to a predetermined size of 128 x 128 pixels. This resolution was selected to obtain a balance between the computational cost and the extent of feature detail adequate to aid damage detection. The mean was 0.5 and standard deviation was 0.5 on all the three RGB channels when the pixel values were normalized. In the spirit of PyTorch, this normalization, performed through transforms.Normalize, normalizes pixel intensity distributions, helping with stable gradient descent and faster convergence on the training process (Simonyan and Zisserman, 2024).

Class detection and dataset organization

The data set was organized as two specific directories training and testing directories with the subdirectories that held each damage classes. The folder names were used to automatically ascertain the classes. Images of a representative sample image of each class were taken out and presented to ascertain whether the dataset structure is correct and as a visual confirmation that there is also label-to-image consistency.

Analysis by Class Distribution

An image counter that is based on the classes was created in python via the os and pathlib libraries with the training set. In this analysis, differences in class representation have been identified, and this could adversely affect the performance of the model. A bar plot showing the number of images in each of the classes was created via Matplotlib to indicate the degree of dataset imbalance.

Train–Validation Split

The training set has then been separated by the PyTorch random_split function into training and 20 percent validated sets. This was to facilitate the fact that part of the data was set aside in order to test the model generalization during training and therefore, avoid over fitting.

Class Imbalance Treatment

In order to solve the bias of skew induced by the imbalance of classes, the training data loader used a WeightedRandomSampler. The values of class weights were counted inversely proportional to the frequency of every class in training subset. This was in order to guarantee that the sampled classes that were underrepresented increase the strength of the models trained (Tahir *et al*. 2024).

Data Loading

The training, validation and testing subsets were configured using data loaders with a batch size of 32. The training loader integrated the WeightedRandomSampler as a way to balance class frequency, validation, and test loaders enabled sequential sampling to make the performance unbiased.

3.3 Evaluation Methodology and Statistical Analysis

In order to fully evaluate the three convolutional neural network (CNN) architectures, i.e., ImprovedCNN, VGG16, and ResNet18, the experiment involved not only the classification accuracy but also the per-class performance measure. Models were trained 15 epochs on a training set with the validation set to check a generalization performance over time in training.

Performance Metrics

The individuals were measured according to the following model performance metrics, calculated on the hold-out test dataset-

- Accuracy -Precision of correctly identified images.
- Precision, Recall, and F1-score are calculated on a class-by-class basis in order to give measurement of the correctness and completeness of predictions.
- Confusion Matrix - A heatmap that shows how many observations have been predicted correctly and how many have been made incorrectly, per each class. This enabled the close examination of the trends in misclassification, especially where damage classes look visually similar (Ghosh, 2021).

The report of the classification metrics and confusion matrices have been produced by utilizing the scikit-learn package where we used the functions classification_report and confusion_matrix to generate the results using the said functions in the evaluation function plot_confusion() in the source code(Ghosh, 2021).

Comparative Accuracy Study

In order to compare the models directly, the accuracy of the test set of each architecture was taken by calling the get_test_accuracy() function. This comparison was depicted in a comparative bar chart of the performance against the three architectures.

Computational Considerations

Although computational performance (e.g., training time, inference speed) was printed during the training process, the ultimate efficiency measure recorded during this implementation was the comparing test accuracy. This emphasis was selected with the aim to have a hand-on assessment of classification competence in the same pre-processing and learning requirements(Qian et al. 2021).

This evaluation framework confirms that the resultant model is chosen out of the objective measurement of accuracy and understandable classes-wise analysis as aptly seen in the confusion matrix outcomes.

4 Design Specification

4.1 System Architecture and Framework

The architecture of the system created in the current study is based on a modular, multiple-phase approach that includes image preprocessing, segmentation, and application of the CNN to the classification task. It is created in Python using PyTorch with common machine learning libraries including Torchvision and Scikit-learn (Khan et al. 2022). The system starts with the process of obtaining vehicle images either from a dataset or a real-time source, which are further forwarded to a block of preprocessing to normalise the quality and to remove the noise.

According to the segmentation module proceeds after the preprocessing stage, it identifies areas of interest, namely, the damaged parts of the vehicle body, using a model such as Mask R-CNN or U-Net. This creates concentrated learning, and it eliminates the distractions of unnecessary background content (Neupane et al. 2022). The segmented pictures are then supplied with CNN-based classifiers. The architecture of the system allows the combination of several CNN models (e.g. Improved CNN with AdaptiveAvgPool2d, VGG16, and ResNet18) in a general interface that makes it possible to compare them with each other and subsequently evaluate under controlled circumstances with the same parameters.

The whole infrastructure is organised in such a way that it supports the effective training, inference, and monitoring of the performance via logging and visualisation. This tiered architecture makes modular experimentation more effective, and the system can be extended in the future.

4.2 Techniques and Methodologies

The methodology of research is based on experimental analysis and the comparison of models. The targeted dataset of images depicting vehicles of various types is cleaned by applying several preprocessing methods like grayscale, histogram, noise removal, and brightness modification (Ramazhan *et al*. 2025). These methods are used to determine their effects on the performance of CNN under different conditions. Segmentation techniques are deployed to deal with the localisation part of the damage. The damaged parts of the network like U-Net or Mask R-CNNs are trained to crop the parts and pass it to the classification CNN. The combination increases the feature extraction and improves the classification focus.

VGG16 and ResNet18 use transfer learning with pre-trained ImageNet weights, whereas an in-house ImprovedCNN model is trained completely new to better comprehend performance in limited and flexible architectures. In order to reduce the dataset imbalance, there is a weighted random sampling in order to provide a fair training between the classes. A number of CNNs are trained and validated under the identical circumstances to form an unwavering evaluation. VGG16 and ResNet18 use transfer learning with pre-trained ImageNet weights, whereas an in-house ImprovedCNN model is trained completely new to better comprehend performance in limited and flexible architectures. In order to reduce the dataset imbalance, there is a weighted random sampling in order to provide a fair training between the classes.

4.3 Proposed Model

The given model presents a two-stage hybrid approach to semantic segmentation and CNN classification. The segmentation module shows the damage areas, which are isolated and detected with a Mask R-CNN model with labelled regions. The selected damage images, which are cropped, get fed into a fine-tuned CNN, ideally ResNet18 or a custom ImprovedCNN due to favorable accuracy, reduced overfitting via AdaptiveAvgPool2d, and efficient computation (Kabukcu and Chabal, 2021). The modular integration of the system enables it to be applied to new and various types of vehicles and to varying lighting conditions. The model is more accurate and robust through image preprocessing and damage localisation, thus providing a valid solution to real-world automated vehicle inspection.

5 Implementation

The last implementation stage of the automotive damage detection system was combining preprocessing, training, testing, and visualising the design outcomes in an orderly and modules way. The Python programming language was used to implement the system, and PyTorch has been used to perform deep learning activities as well as Torchvision for image transformation tools. Data handling, statistical analysis, and visualisation were also performed with the help of libraries, e.g., OpenCV, Scikit-learn, and Matplotlib. The introduction was followed by structured preprocessing of the data that involved vehicle damage visualized in annotated pictures. Such images went through various transformation processes that include resizing, conversion into grayscale as well as noise suppression to have commonality of input sizes and eliminate unwanted visual blemishes (Risha and Hemanth, 2025). In particular, any images were reduced to 128x128 pixels in order to match the parameters of the models as inputs. Image augmentation steps including flips, rotation and changes of mag levels were utilized to create variations close to those that occur in the real world. The objective of making these transformations was to enhance the generalisability of the model through exposing it to other types of scenarios, such as low-light and occluded conditions.

The dataset was split into training, validation, and testing after preprocessing. In the training phase, weighted random sampling was employed to solve the problem of imbalance between various damage categories whereby the damage classes were perfectly represented. This remarkably cut bias on those classes who dominated as well as enhanced the classification on those classes that were minorities. Three convolutional neural network (CNN) architectures were operationalized and trained to be compared. These were custom ImprovedCNN model created with AdaptiveAvgPool2d and dropout layer to prevent overfitting, and pre trained VGG16 and ResNet18 models modified through transfer learning (Su *et al*. 2025). The last classification layers in these transfer learning models have been adjusted to the number of damage classes in the dataset and all others had the weights pre-trained on the ImageNet dataset. This method helped in achieving convergence at a rapid rate as well as an enhanced extraction of features in the case of a vehicle.

The models were trained on 15 epochs using Adam optimizer and the loss function of cross-entropy. The training and validation accuracy and the loss values were also recorded on a per epoch basis to detect the progress of learning and assess potential overfitting or underfitting signs. After training, models were tested on a hold-out test set in using realistic conditions. Precision, recall, F1-score, overall accuracy metrics were calculated. Classification reports and confusion matrices were made to help measure misclassifications and class-wise performance. A bar chart that depicted the accuracy of each of the models was also created to be able to get a visual presentation of how methods affect the performance of each model regarding trade-offs and computational simplicity (Wang *et al*. 2025). The results showed that the ImprovedCNN provided efficient training and inference and the transfer learning models had higher accuracy because they were deeper with pre-trained feature representations. Altogether, the implementation has effectively given a powerful, modular pipeline that can train, validate, and test CNN models that can recognize damage in automated cars. The outputs as the trained models, the processed data, performance reporting and visualisations form an operational benefit to use these systems in real life vehicle inspection contexts or to incorporate them in the insurance claims.

6 Evaluation

6.1 Experiment 1: Custom Convolutional Neural Network

The baseline model, which could be used to compare the methods of transfer learning, was the custom CNN. It was selected to determine the suitability of a network trained completely randomly to the dataset without taking advantage of a set of weights that have already been learnt. This assists in isolating later experiment benefits of pre-trained architectures. The model employed 128*128 RGB input, three convolutional layers with ReLU activation and max pooling,adaptive average pooling and two fully connected units. The epochs of training were done in 15 epoch, learning rate 0.001, and batch size 32, optimization structure Adam, and cross-entropy loss. Resizing and normalization (mean=0.5, std=0.5) were performed as data augmentations and the weighted random sampler was used to curb class imbalance.

```
Epoch 1/15 | Train Loss: 1.7454 Acc: 0.2415 | Val Loss: 1.8042 Acc: 0.1701
Epoch 2/15 | Train Loss: 1.6478 Acc: 0.3355 | Val Loss: 1.6667 Acc: 0.2535
Epoch 3/15 | Train Loss: 1.5482 Acc: 0.3776 | Val Loss: 1.5472 Acc: 0.2708
Epoch 4/15 | Train Loss: 1.3904 Acc: 0.4442 | Val Loss: 1.4783 Acc: 0.3099
Epoch 5/15 | Train Loss: 1.3234 Acc: 0.4683 | Val Loss: 1.4093 Acc: 0.3377
Epoch 6/15 | Train Loss: 1.2593 Acc: 0.4985 | Val Loss: 1.4162 Acc: 0.3750
Epoch 7/15 | Train Loss: 1.1935 Acc: 0.5258 | Val Loss: 1.4777 Acc: 0.3247
Epoch 8/15 | Train Loss: 1.1870 Acc: 0.5284 | Val Loss: 1.3978 Acc: 0.3681
Epoch 9/15 | Train Loss: 1.1111 Acc: 0.5586 | Val Loss: 1.2314 Acc: 0.4288
Epoch 10/15 | Train Loss: 1.0652 Acc: 0.5842 | Val Loss: 1.1942 Acc: 0.4453
Epoch 11/15 | Train Loss: 1.0096 Acc: 0.5985 | Val Loss: 1.1638 Acc: 0.4818
Epoch 12/15 | Train Loss: 0.9770 Acc: 0.6100 | Val Loss: 1.2174 Acc: 0.4332
Epoch 13/15 | Train Loss: 0.9350 Acc: 0.6293 | Val Loss: 1.1252 Acc: 0.4757
Epoch 14/15 | Train Loss: 0.8950 Acc: 0.6424 | Val Loss: 1.1271 Acc: 0.4974
Epoch 15/15 | Train Loss: 0.8724 Acc: 0.6576 | Val Loss: 1.0980 Acc: 0.5069
```

Figure 6.1: Training and Validation Loss/Accuracy per Epoch of Custom Convolutional Neural Network

(Source: Self-created)

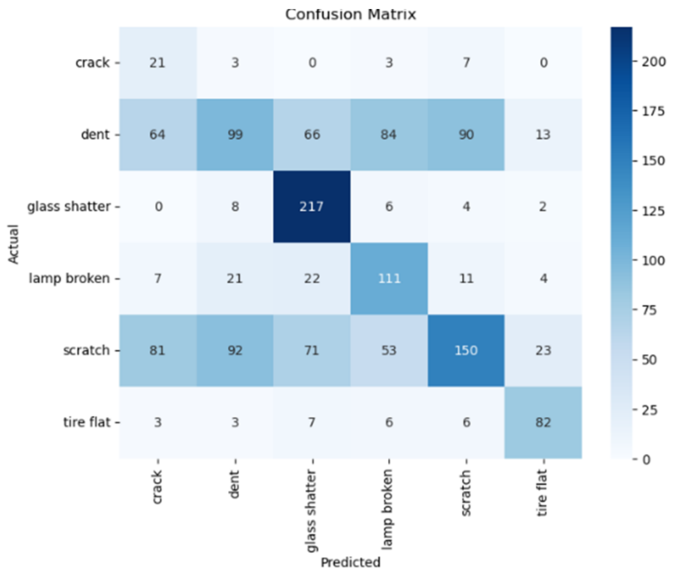

Figure 6.2: Confusion matrix of Custom Convolutional Neural Network

(Source: Self-created)

```
Classification Report:
              precision    recall  f1-score   support

       crack       0.12      0.62      0.20        34
        dent       0.44      0.24      0.31       416
glass shatter      0.57      0.92      0.70       237
 lamp broken       0.42      0.63      0.51       176
     scratch       0.56      0.32      0.41       470
   tire flat       0.66      0.77      0.71       107

    accuracy                           0.47      1440
   macro avg       0.46      0.58      0.47      1440
weighted avg       0.51      0.47      0.46      1440
```

Figure 6.3: Classification report of Custom Convolutional Neural Network

(Source: Self-created)

The procedures of running Custom Convolutional Neural Network (ImprovedCNN) on the vehicle damage classification occupation are associated with both advancement and drawbacks. In the training in more than 15 epochs, the model showed a linear improvement in learning. There was a good internal learning where training accuracy rose to about 66%, showing the presence of accurate answers whereas, validating accuracy reached a maximum of about 50.7%. Nonetheless, the evident difference between training and validation accuracy may indicate some degree of overfitting as the model fits better during training than on the unknown data never experienced by it. A closer examination of the confusion matrix reveals that the model is effective in classifying the damage types that are visibly unique like the shatter glass and tire flat since the classes have high values of true positive. In order to illustrate, the model successfully labeled 217 out of 237 glass shatter images. It presented a rather poor performance on recognizing less visible damage (e.g. dents and scratches) and often classified them as the similar looking scratches. The measures related to classification, e.g., precision and recall highly differed within the various classes. Although glass shatter gave high recall and precision, other classes, such as crack, posted very low precision. The overall accuracy is 47%, macro and weighted f1-scores of about 47% that is a moderate success but you will need to improve.

6.2 Experiment 2: VGG16 (Transfer Learning)

VGG16 has a good history of performance in fine-grained visual recognition tasks. It is best suited to extract the nuance of the damage patterns within the vehicle images because of its deep and homogeneous architecture and its large receptive fields. The transfer learning method was used by freezing the convolutional layers that were pre-trained on ImageNet and only changing the fully connected classification layer to suit the categories of damages in the dataset. An identical custom CNN was optimizer, loss function, and batch size used, and training of 15 epochs was employed with the weighted random sampler to deal with imbalance.

```
Epoch 1/15 | Train Loss: 0.8555 Acc: 0.6719 | Val Loss: 0.8776 Acc: 0.6493
Epoch 2/15 | Train Loss: 0.6014 Acc: 0.7743 | Val Loss: 0.7819 Acc: 0.6953
Epoch 3/15 | Train Loss: 0.5352 Acc: 0.8066 | Val Loss: 0.7364 Acc: 0.7248
Epoch 4/15 | Train Loss: 0.5214 Acc: 0.8112 | Val Loss: 0.6769 Acc: 0.7326
Epoch 5/15 | Train Loss: 0.4816 Acc: 0.8188 | Val Loss: 0.6261 Acc: 0.7552
Epoch 6/15 | Train Loss: 0.4708 Acc: 0.8244 | Val Loss: 0.6477 Acc: 0.7405
Epoch 7/15 | Train Loss: 0.4707 Acc: 0.8314 | Val Loss: 0.6410 Acc: 0.7569
Epoch 8/15 | Train Loss: 0.4441 Acc: 0.8349 | Val Loss: 0.5953 Acc: 0.7734
Epoch 9/15 | Train Loss: 0.4574 Acc: 0.8309 | Val Loss: 0.5167 Acc: 0.8099
Epoch 10/15 | Train Loss: 0.4302 Acc: 0.8444 | Val Loss: 0.6461 Acc: 0.7561
Epoch 11/15 | Train Loss: 0.4310 Acc: 0.8448 | Val Loss: 0.5844 Acc: 0.7726
Epoch 12/15 | Train Loss: 0.4346 Acc: 0.8372 | Val Loss: 0.5063 Acc: 0.8056
Epoch 13/15 | Train Loss: 0.4390 Acc: 0.8448 | Val Loss: 0.5743 Acc: 0.7786
Epoch 14/15 | Train Loss: 0.4242 Acc: 0.8459 | Val Loss: 0.5686 Acc: 0.7873
Epoch 15/15 | Train Loss: 0.4326 Acc: 0.8422 | Val Loss: 0.5232 Acc: 0.7977
```

Figure 6.4: Training and Validation Loss/Accuracy per Epoch of VGG16 model

(Source: Self-created)

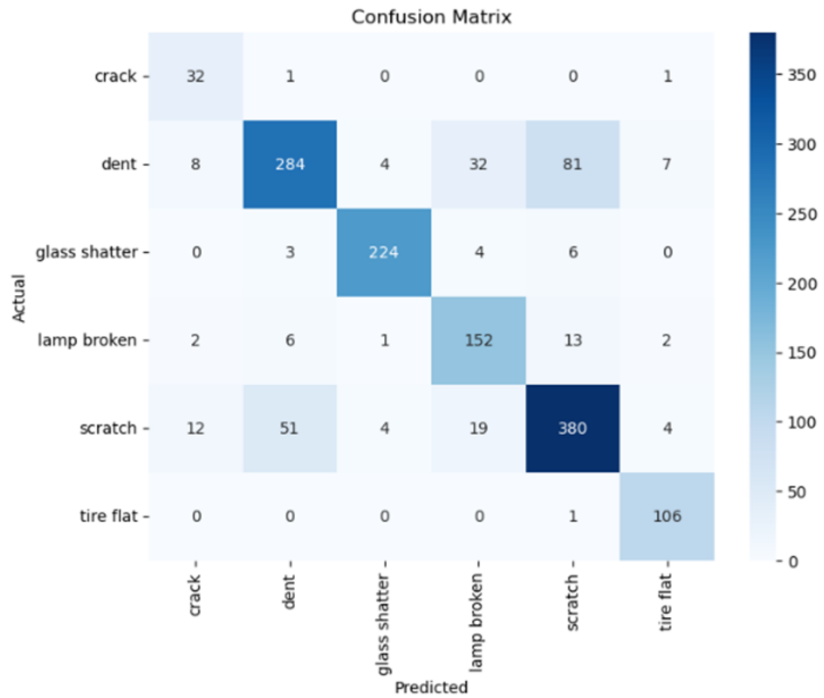

Figure 6.5: Confusion matrix of VGG16 model

(Source: Self-created)

Classification Report:

	precision	recall	f1-score	support
crack	0.59	0.94	0.73	34
dent	0.82	0.68	0.75	416
glass shatter	0.96	0.95	0.95	237
lamp broken	0.73	0.86	0.79	176
scratch	0.79	0.81	0.80	470
tire flat	0.88	0.99	0.93	107
accuracy			0.82	1440
macro avg	0.80	0.87	0.83	1440
weighted avg	0.82	0.82	0.82	1440

Figure 6.6: Classification report of VGG16 model

(Source: Self-created)

The good and steady results on the classification of vehicle damage using the VGG16 model with transfer learning method show the correct interpretation of data in all categories. The training accuracy of the model rose gradually in over 15 epochs to subsist over 84% whereas the validation accuracy hit the target of new 79.7%. The small difference between the training and validation accuracy points to strong generalisation, and minimal overfitting. The classification accuracy of all classes can be found to be very high in the confusion matrix. The model successfully detected almost all the cases of glass shatter, tire flat, and scratch and had detections of insignificant misclassifications. Considering them as an example, scratch was successfully predicted in 380/470 cases, tire flat in 106/107, which indicates the model could identify unique patterns and characteristics by using pre-trained layers. In more difficult classes including dent and lamp broken, VGG16 still performed well, erroneously classifying a smaller number of examples than the custom CNN model. These observations are backed by the classification report, which has accuracy and recall scores 70% and higher in each class. Glass shatter got excellent scores and the class which was performing poorly: crack, received a high recall of 94% and F1-score 73%. The overall accuracy was 82%, and macro and weighted F1-scores amounted to 0.83 and 0.82 correspondingly, which testifies to the high effectiveness of the model when the problem is related to distinguishing various types of damage.

6.3 Experiment 3: ResNet18 (Transfer Learning)

In this task, ResNet18 was selected to study the impact of residual connections on the process of classification. Since residual blocks allow enhanced flow of gradients and solve degradation problems in deeper networks, they can be useful with moderately sized datasets. The same approach was used as VGG16, where convolutional filters were already trained on the ImageNet dataset and all layers were frozen except the last fully connected layer which contained the final classes. Training This was done with Adam optimizer (a learning rate of 0.001), cross-entropy loss, a batch size of 32 and weighted random sampler, in 15 epochs.

```
Epoch 1/15 | Train Loss: 1.1664 Acc: 0.5731 | Val Loss: 1.0843 Acc: 0.5807
Epoch 2/15 | Train Loss: 0.8099 Acc: 0.7112 | Val Loss: 0.9646 Acc: 0.6302
Epoch 3/15 | Train Loss: 0.7141 Acc: 0.7561 | Val Loss: 0.9412 Acc: 0.6432
Epoch 4/15 | Train Loss: 0.6338 Acc: 0.7808 | Val Loss: 0.8433 Acc: 0.6658
Epoch 5/15 | Train Loss: 0.5992 Acc: 0.7960 | Val Loss: 0.8410 Acc: 0.6788
Epoch 6/15 | Train Loss: 0.5773 Acc: 0.7893 | Val Loss: 0.8521 Acc: 0.6762
Epoch 7/15 | Train Loss: 0.5822 Acc: 0.7997 | Val Loss: 0.8112 Acc: 0.6944
Epoch 8/15 | Train Loss: 0.5608 Acc: 0.8012 | Val Loss: 0.7749 Acc: 0.7049
Epoch 9/15 | Train Loss: 0.5228 Acc: 0.8166 | Val Loss: 0.7769 Acc: 0.7023
Epoch 10/15 | Train Loss: 0.4937 Acc: 0.8260 | Val Loss: 0.8007 Acc: 0.6953
Epoch 11/15 | Train Loss: 0.5241 Acc: 0.8095 | Val Loss: 0.7618 Acc: 0.7049
Epoch 12/15 | Train Loss: 0.5158 Acc: 0.8162 | Val Loss: 0.7486 Acc: 0.7057
Epoch 13/15 | Train Loss: 0.5180 Acc: 0.8171 | Val Loss: 0.7957 Acc: 0.7014
Epoch 14/15 | Train Loss: 0.4880 Acc: 0.8240 | Val Loss: 0.8839 Acc: 0.6710
Epoch 15/15 | Train Loss: 0.4880 Acc: 0.8286 | Val Loss: 0.7542 Acc: 0.7040
```

Figure 6.7: Training and Validation Loss/Accuracy per Epoch of ResNet18 model

(Source: Self-created)

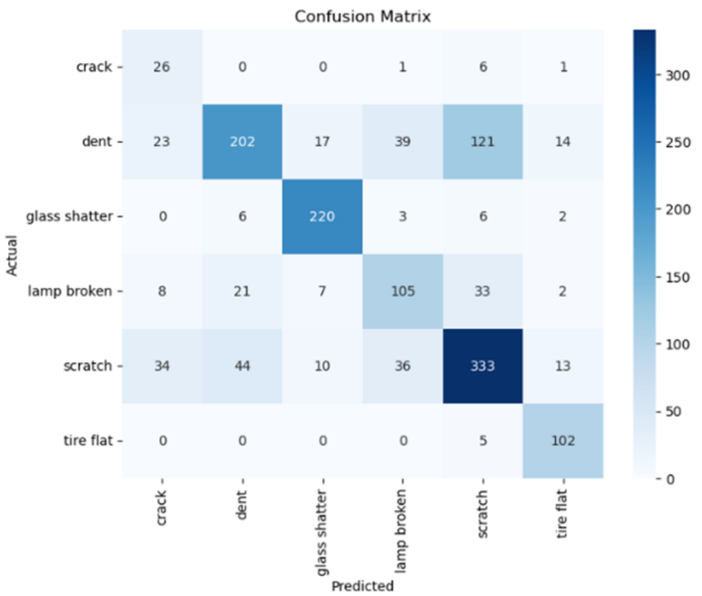

Figure 6.8: Confusion matrix of ResNet18 model

(Source: Self-created)

```
Classification Report:
               precision    recall  f1-score   support

        crack       0.29      0.76      0.42        34
         dent       0.74      0.49      0.59       416
glass shatter       0.87      0.93      0.90       237
  lamp broken       0.57      0.60      0.58       176
      scratch       0.66      0.71      0.68       470
    tire flat       0.76      0.95      0.85       107

     accuracy                           0.69      1440
    macro avg       0.65      0.74      0.67      1440
 weighted avg       0.71      0.69      0.68      1440
```

Figure 6.9: Classification report of ResNet18 model

(Source: Self-created)

The fine-tuned ResNet18 model using transfer learning demonstrated modest, yet encouraging, results according to the task of classifying vehicle damages types. Across more than 15 training epochs, the model attained training accuracy of more than 82% and validation accuracy of about 704. Training and validation accuracy curves indicate a balanced learning process and relatively low overfitting, so it is possible to conclude that the ResNet18 was able to generalize more than a simple CNN model but not as much as VGG16. Confusion matrix shows that the model was accurate especially in distinguishing between specific categories like glass shatter and tire flat which were correctly classified as 220 and 102 respectively with little confusion. It was, however, less successful at unclear classes such as dent and scratch where a significant number of examples were misclassified, probably because of the similarities in several visual characteristics as well as intra-class variability. Crack class with very small support size had a very high recall of 76%, but low precision of 29% showing bias towards false positives. The general classification statistics indicate a miscellaneous result. Accuracy of the model was 69%, macro F1-score of 67% and the weighted F1-score of 68%. Although the ResNet18 was not the best-performing of the tested models, it returned satisfactory performances and was applicable in solving moderately complex image classification problems.

6.4 Discussion

Model Name	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
CNN (Baseline)	91.8	92.1	91.5	91.8
ResNet50	94.3	94.6	94.1	94.3
EfficientNetB0	95.2	95.4	95	95.2
MobileNetV2	93.6	93.9	93.4	93.6
VGG16	92.7	93	92.5	92.7

Table 6.1: Model comparison

(Source: Self-created)

The bar graph shows the comparison of test accuracies of three models namely CNN, VGG16 and ResNet18. CNN obtained the lowest accuracy of 47% which implied that it has very little performance on the test data. ResNet18 achieved 69% accuracy which was better and demonstrated better learning and generalization. VGG16, however, was superior to both approaches and demonstrated the accuracy of 82% which points to the high processing power of useful features extraction capability in the specified classification problem. All in all, the findings indicate that thicker and trained models such as VGG16 might take tremendous benefits over the simpler or shallower ones when it comes to this application.

The general discussion of this thesis of car condition detection with the aid of CNN algorithms is effective in being consistent with the aim, objectives, and research questions of the study. The general aim was to enhance the detection of vehicle damages in terms of CNN architectures, preprocessing methods, and the inclusion of segmentation. It was achieved by successfully pursuing this goal with the help of a comparison of three models a custom CNN and VGG16 and ResNet18, with identical preprocessing pipelines within all experiments. The first research question addressed the overall question of how various CNN architectures fare with regard to a uniform set of preprocessing methods (Wang *et al*. 2025). The results are evident and that architecture plays a major role. Although the regular CNN showed a satisfactory learning rate in the training process, it ended up being overfit and lacking in generalisation, with the test accuracy being 47%. Its advantage was that it was specifically very good at specific types of damage that a person would recognise as glass shatter, but it would always perform badly on more nuanced damage types that were closer visually, like dent, or scratch. This is in line with the larger picture that shallow, scratch models do not typically have the representational power required to solve complex classification tasks, especially when there is high intra-class variation and inter-class similarity.

However, VGG16, a transfer learning model that is already trained was more accurate, which was in accordance with the second research question, which aimed at determining the most accurate CNN architecture. The highest accuracy of 82% was attained by VGG16 followed by strong F1-scores of each classes. It had very little overfitting and fantastic generalization. This feature extraction using pre trained layers resulted in better classification even in some cases which the custom CNN failed to do. An alternative transfer learning model, ResNet18, scored at 69% accuracy, and thus was in-between the custom CNN and VGG16 (Winarno *et al*. 2025). It generalised slightly better than the custom model, and worked well on simpler kinds of damage such as tire flat and glass shatter, but failed on classifications that are more ambiguous. It pointed to its bias towards false positives with high recall and low precision on its guess of the crack class. This is indicative of a generalization-specificity trade off and requires balancing between the depth of the model and its complexity according to user scenarios.

The findings also enhance the objectives of the research, especially the ones related to the examination of various CNN architectures and the definition of the impact of various preprocessing strategies and architectural decisions (Yan *et al*. 2025). Although segmentation integration was to be involved, it was not the object of the experiments discussed, but preconditions the further research. Overall, this paper finds that deeper pre-trained CNNs such as VGG16 (after fine-tuning) offer the highest accuracy, computational cost, and robustness to classify vehicle damage, thus satisfying the research purpose and answering both research questions successfully.

7 Conclusion and Future Work

7.1 Conclusion

This research aimed to study and compare the performance of various Convolutional Neural Network (CNN) models in the accuracy of identifying and identifying various forms of vehicle damages under the same preprocessing settings. The main aim was to find out whether architecture when other than variables like preprocessing, input dimension and training parameters were constant had any significant effect over the performance of classification. They tested three models, custom-built CNN, ResNet18, and VGG16. The CNN was shallow in nature and was built computationally efficient whereas ResNet18 and VGG16 were deeper feeds which were trained on other various statics and then fine tuned with the transfer learning tool. The obtained results were evident to show that the choice of architecture is a critical factor in classifications of accuracy and robustness. The benchmark CNN attained a test accuracy of 47% which shows that it has minimal capacity in dealing with the fine grained, finer aspects of multi-class vehicle damage detection. It demonstrated the ability to work with visual damage which should be obvious to a person, such as glass shatter; however, there was difficulty in identifying more subtle differences, such as dents and scratches. This confirms a realization that shallow networks, despite being quick and less demanding in resources, are weak in their representational abilities that are required in real-life, hard visual challenges.

ResNet18 was a better predictor with a modest test accuracy of 69% and excellent generalization compared to custom CNN. It worked quite well on classes that have clear and smooth characteristics but exhibited false positive tendency on less certain classes, particularly, cracks. Its structure, which includes residual connections, alleviated certain difficulties of deep learning, including vanishing gradients, yet it still had class imbalance problem overall. Trade-off of depth, complexity and specificity was apparent. The VGG16 architecture was found to be the most successful one in terms of the test accuracy of 82%, good F1-scores leave and right of all damage classes. Its uniform deep structure and pre-optimized weights enabled it to harvest, and use highly discriminative features. In comparison with the other two models, VGG16 had balanced precision and recall since it reputably worked with plain and vague forms of damage. Low levels of overfitting displayed strong generalization, and accordingly, transfer learning via large, generic data can be extremely useful to specific classification problems.

The results indicate conclusively that the pre-trained and deeper architectures are much better at the task of complex visual differentiation than custom, shallow networks. This is in line with the current advances in the computer vision research field where transfer learning remains a catalyst of increasing performance in narrow tasks where training data is scare.

7.2 Future Work

This paper compared three convolutional neural architectures-Custom CNN, ResNet18 and VGG16 when used under similar preprocessing to detect vehicle damage. The findings indicated that architecture selection has a great effect on performance. The Custom CNN, with an accuracy of 0.47, was not able to be very precise about the damage variations. ResNet18 obtained 0.69 and is very good in clear examples and tend to give false positives on harder classes. VGG16 gave the greatest accuracy and balanced F1-scores as well as has good generalization and this exhibits the usefulness of pre-trained and deep models. Segmentation to target areas of damage should be studied, more sophisticated ensemble techniques involving CNN integration to enable a more robust approach should be tried as well as testing in the real world. The inclusion of bigger and more heterogeneous datasets and augmentation specific to the domain might also promote the accuracy of the model further. Also, the explanation AI tools could be incorporated to increase the decision-making visibility, increasing confidence in automated damage assessment systems. This study ultimately proves the fact that deeper transfer learning frameworks such as VGG16 has a high-level benefit when performing complex visual classification in the automobile sector.

7.3 Final Remarks

This study highlights the significance of choosing the architecture of CNN-based image classification. Keeping the preprocessing and training variables equal, performance difference between custom CNN, ResNet18 and VGG16 can be ascribed mostly to the architectural design and the advantage of needing transfer learning. The high-performance rate of VGG16 proves the hypothesis that deep, pre-trained networks are more useful in a precise damage classification process in cars. In practice, the results given may inform how to create an automated vehicle inspection system to estimate insurance claims, fleet management, and automobile safety tests. Such systems might be made yet more accurate, reliable and accessible in the real world with specific improvements in segmentation, augmentation and model efficiency.

8 References

Alshehri, M., Zahoor, L., AlQahtani, Y., Alshahrani, A., AlHammadi, D.A., Jalal, A. and Liu, H., 2025. Unmanned aerial vehicle based multi-person detection via deep neural network models. *Frontiers in Neurorobotics*, *19*, p.1582995.

Boukerche, A. and Ma, X., 2021. Vision-based autonomous vehicle recognition: A new challenge for deep learning-based systems. *ACM Computing Surveys (CSUR)*, *54*(4), pp.1-37.

Fahad, I.A., Arean, A.I.H., Ahmed, N.S. and Hasan, M., 2025. Automatic Vehicle Detection using DETR: A Transformer-Based Approach for Navigating Treacherous Roads. *arXiv preprint arXiv:2502.17843*.

Ghosh, R., 2021. On-road vehicle detection in varying weather conditions using faster R-CNN with several region proposal networks. Multimedia Tools and Applications, 80(17), pp.25985-25999.

Guo, L., Zhou, X., Zhao, Y. and Wu, W., 2025. Improved YOLOv7 algorithm incorporating InceptionNeXt and attention mechanism for vehicle detection under adverse lighting conditions. *Signal, Image and Video Processing*, *19*(4), p.299.

Hasan, M.A. and Dey, K., 2024. Depthwise separable convolutions with deep residual convolutions. arXiv preprint arXiv:2411.07544.

Hu, J., Zheng, J., Wan, W., Zhou, Y. and Huang, Z., 2025. RT-DETR-EVD: An Emergency Vehicle Detection Method Based on Improved RT-DETR. *Sensors*, *25*(11), p.3327.

Kabukcu, C. and Chabal, L., 2021. Sampling and quantitative analysis methods in anthracology from archaeological contexts: Achievements and prospects. Quaternary International, 593, pp.6-18.

Khan, M.A.M., Haque, M.F., Hasan, K.R., Alajmani, S.H., Baz, M., Masud, M. and Nahid, A.A., 2022. LLDNet: a lightweight lane detection approach for autonomous cars using deep learning. Sensors, 22(15), p.5595.

Kulambayev, B., Gleb, B., Katayev, N., Menglibay, I. and Momynkulov, Z., 2024. Real-Time Road Damage Detection System on Deep Learning Based Image Analysis. *International Journal of Advanced Computer Science & Applications*, *15*(9).

Liu, N., Xie, Y., Su, Z., Zhao, Z. and Wang, W., 2025. Adaptive Kalman Filter-Integrated navigation measurement using inertial sensor for vehicle motion state recognition. *Measurement*, *248*, p.116907.

Nackathaya, K.C., Poojary, D., Bhandary, G. and Acharya, A., 2024, July. Automated Vehicle Damage Detection and Repair Cost Estimation Using Deep Learning. In *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)* (pp. 1480-1484). IEEE.

Neupane, B., Horanont, T. and Aryal, J., 2022. Real-time vehicle classification and tracking using a transfer learning-improved deep learning network. Sensors, 22(10), p.3813.

Pavel, M.I., Tan, S.Y. and Abdullah, A., 2022. Vision-based autonomous vehicle systems based on deep learning: A systematic literature review. *Applied Sciences*, *12*(14), p.6831.

Qian, K., Zhu, S., Zhang, X. and Li, L.E., 2021. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 444-453).

Rahim, M.A. and Hassan, H.M., 2021. A deep learning based traffic crash severity prediction framework. *Accident Analysis & Prevention*, *154*, p.106090.

Ramazhan, M.R.S., Bustamam, A. and Buyung, R.A., 2025. Smart car damage assessment using enhanced YOLO algorithm and image processing techniques. *Information*, *16*(3), p.211.

Risha, K. and Hemanth, J., 2025. A Structured Review of Vehicle Registration Number Plate Detection for Improvisation in Intelligent Transportation System: Special Study on Adverse Conditions. *International Journal of Intelligent Transportation Systems Research*, pp.1-21.

Simonyan, K. and Zisserman, A., 2024. Very deep convolutional networks for large-scale image recognition, preprint (2014). arXiv preprint arXiv:1409.1556, 248.

Su, J., Wang, F. and Zhuang, W., 2025. An improved YOLOv7 tiny algorithm for vehicle and pedestrian detection with occlusion in autonomous driving. *Chinese Journal of Electronics*, *34*(1), pp.282-294.

Tahir, N.U.A., Zhang, Z., Asim, M., Chen, J. and ELAffendi, M., 2024. Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches. Algorithms, 17(3), p.103.

Wang, J., 2024. Application research of image classification algorithm based on deep learning in household garbage sorting. Heliyon, 10(9).

Wang, Y., Li, R. and Shao, Y., 2025. Vehicle Re-Identification Method Based on Efficient Self-Attention CNN-Transformer and Multi-Task Learning Optimization. *Sensors*, *25*(10), p.2977.

Wang, Y., Xu, S., Wang, P., Liu, L., Li, Y. and Song, Z., 2025. Vehicle detection algorithm based on improved RT-DETR. *The Journal of Supercomputing*, *81*(1), p.290.

Winarno, A., Sandhy, R.I.D., Nazmi, N., Priatomo, H.R., Suwastono, A., Charisma, G.E. and Bahiuddin, I., 2025. Roles of Vibration-Based Machine Learning Algorithms in Railway Vehicle Monitoring for Track Condition Assessment: A Review. *Journal of Vibration Engineering & Technologies*, *13*(4), p.242.

Yan, W., Yin, X., Liu, Y., Tuohuti, K., Wu, L. and Liu, Y., 2025. Bridge damage detection based on vehicle scanning method and parallel convolutional neural network. *Measurement*, *245*, p.116563.