Damage Detection in Vehicles

By

Austin Vornhagen

Department of Computer Science

College of Information Science and Technology

University of Nebraska at Omaha

Omaha, NE 68182-0500

austinvornhagen@gmail.com

For

8300 - Image Processing and Computer Vision

Fall 2023

Thursday, December 7, 2023

Abstract

Utilizing supervised learning and computer vision, this research develops an innovative method for automated vehicle damage assessment that can infer interior damage from exterior damage imagery. The core innovation of this method is its unique predictive capability, which allows for the estimation of unseen interior damage—a feature not currently available in photo-based estimation systems. This advancement refines existing damage assessment models and contributes new perspectives on the relationship between exterior and interior damage, significant for the automotive engineering field. The findings demonstrate the system's potential to revolutionize damage assessment practices, offering a pathway to more precise and dependable repair cost estimations.

Table of Contents

I. Introduction 5

I.1. Problem 5

I.2. Motivations 5

I.3. Significance 5

I.4. Challenges 6

I.5. Objectives 6

II. Overview 7

II.1. History of the problem 7

II.2 State of the art 8

III. Techniques 10

III.1. Principles, Concepts, and Theoretical Foundations of the research problem 10

III.2. Techniques that have been used by other researchers for the research problem 11

III.3. Relevant technologies that would be useful to this research 14

III.4. Algorithms, Process modules, and solution methods 15

IV. Approaches 15

IV.1. Methodologies I applied in this research 15

IV.2. Techniques I used to solve the problem 15

IV.3. Processes I engaged in this research 17

IV.4. Facilities and supplies used for this research 18

V. Workplan 18

V.1. Tasks performed in this research 18

V.2. Schedule, timeline, and milestones 20

VI. Results and Analysis 21

VI.1. Results 21

VI.2. Analysis of the results 21

VI.3. Discussions 22
VI.3.1. Problems and issues arisen during the research 22
VI.3.2. Limitation and constraints of the research (and results) 23
VII. Summary 23
References 26
Appendix 27
Dataset 27
Annotation 36
Ontology 41
Model Design 46
Model Training 49

I. Introduction

I.1. Problem

The core problem studied in this research is the accurate detection and quantification of car damage using computer vision techniques.

I.2. Motivations

Manual inspections can be time-consuming, subjective, and potentially inaccurate. Automated inspections only consider exterior damage. Cars are frequently exposed to various external factors that induce wear, tear, or damage. This problem needs to be studied because an improved automated damage detection and quantification process provides many benefits including ensuring all safety concerns are addressed before the vehicle is deemed roadworthy again, making informed decisions about whether a car should be repaired or be replaced, speeding up insurance claims, increased accuracy, increased consistency, preventing overcharging or underestimating costs of repairs, preventing fraud where the extent of damage is exaggerated, increased customer trust, technological advancement, efficient resource utilization, could create new jobs, increased accessibility in remote locations, etc.

I.3. Significance

Quantifying damage on cars can be challenging. Overcoming these challenges can advance science and technology and push the boundaries of current computer vision understanding leading to innovations in image processing techniques. As more damage is detected and categorized, there will be a growing repository of data. This can be beneficial for advancing the solutions to this problem.

The problem is not yet solved because the automated estimation systems do not yet infer interior damage from pictures of exterior damage. Quantifying damage to cars depends on material damage, damage location, damage severity, material available, labor available, cause of damage, environmental conditions, etc. This variability makes it challenging to find a solution.

The solution in this paper is worth considering because the world can benefit from improvements in estimating the cost of damage to cars. Many people with car insurance and insurance companies can benefit. The methods and approach should inspire someone else to build off the presented solution. This will be more effective than current methods because an automated system can remain objective and it will infer the damage to the interior of the car in addition to the damage to the exterior of the car.

I.4. Challenges

Damage detection and quantification of repair costs require a system that understands wear, tear, and impact damage. It is not a binary classification problem but requires multiple nuanced, graded evaluations. While there have been attempts to use computer vision in damage detection and quantification, the added dimension of inferring damage costs of the interior and the integration of it into the main system makes the problem unique. Existing solutions might detect a dent but providing an accurate assessment of its depth, area, and impact on structural integrity beneath the dent is still a largely uncharted domain. With the advancements in deep learning and image processing, this problem is within the realm of solvability.

Damage can range from minor scratches to significant dents or structural flaws. Training a model to recognize this vast spectrum is challenging. Shadows, lighting conditions, reflections, or weather conditions can impact the visibility and discernibility of the damage, complicating the detection process. It is difficult to put together a comprehensive dataset that covers the many types of potential damage. Determining the depth and severity of the damage is also difficult. However, the hardest part is what makes this research novel, detecting certain kinds of damage like engine damage or damage to other internal components from pictures.

I.5. Objectives

The goals of the research are: to create a new ontology for detecting and classifying car damage, create a model that detects and quantifies car damage, test the model across different car models and types, and successfully infer damage to the interior components of the car from images of exterior damage as well as including the interior damage in the cost.

It is expected that a prototype system will be developed offering rapid and precise damage assessments. This would hopefully reduce the time taken to evaluate damage and lead to cost savings in repair and maintenance processes.

Hypothesis: The model will achieve a detection accuracy rate of 50% on interior car damage and a quantification accuracy rate of 50% on exterior car damage.

II. Overview

II.1. History of the problem

As society used more vehicles, there was an increasing need for consistent quantification of damage specifically from people with car insurance. Initially, car damage was assessed manually by trained professionals. The problem arises from the need to assess car damage quickly, efficiently, and objectively. Manual inspections might overlook or inaccurately gauge damage and a qualified individual needs to be present which sometimes is not available. As technology evolved, there were attempts to automate the process using simple imaging tools.

Early solutions have utilized basic photo comparisons or rudimentary software to highlight differences between an undamaged reference image and the current state. Recent approaches have delved into neural networks and sophisticated image-processing techniques to detect vehicle damage and quantify repair costs. Numerous researchers, engineers, and professionals spanning the fields of automotive engineering, computer science, and insurance have worked on this problem.

(Patil et al. 2017) employed a Convolutional Neural Network for car damage classification into the categories: bumper dent, door dent, glass shatter, headlamp broken, tail lamp broken, scratch, and smash. At the time, there were not any public datasets for car damage classification so they created their own dataset by scraping the web. They achieved 88.24% accuracy with individual models and 89.53% accuracy using ensemble methods.

(Kyu et al. 2020) applied VGG16 and VGG19 for car damage detection and assessment in real-world datasets. They achieved an accuracy of 95.22% of VGG19 and 94.56% of VGG16 in damage detection, 76.48% VGG19 and 74.39% VGG16 in damage localization, and 58.48% VGG19 and 54.8% VGG16 in damage severity with a combination of transfer learning and L2 regularization.

(Zhang et al. 2020) conducted experiments comparing damage assessment accuracy between acquiring videos and images with different apps. The average accuracy of videos is 29.1% higher and the ratio of high-quality shooting data on predefined criterion is also 20% higher.

Raap, M. (2021) investigated using semi-supervised learning. One benefit is if successful is to reduce the amount of work needed when creating a vehicle damage detection dataset. The goal was to see if a semi-supervised learning model could achieve similar or even better vehicle damage detection accuracy rates as fully supervised models. The research in this paper was not able to show Convolution Neural Network models enriched with enhanced pseudo-labelling methods achieve a statistically higher (with 95% confidence) precision and recall in a semi-supervised setting than simple Convolution Neural Network models when utilized to detect the damage on vehicles. Fully supervised achieved 84.37% accuracy and semi supervised achieved 81.28% accuracy.

II.2 State of the art

This problem currently is faced by the insurance industry. Car insurance is required in all states except Florida. When a car receives damage, they can use a photo estimate, or take the car to a repair shop and have a trained professional give an estimate. Insurer's will expect you to get at least one estimate. The estimate documents both the repairs that will be performed, and what they are expected to cost. Then the estimate through your insurance carrier could take 2-5 days to process and be approved. Sometimes, because photo estimates can only see the outside of the car, the cost of repairs is underestimated. When the auto insurance estimate is too low, you can file a supplemental claim to reimburse your costs, but you still must pay the difference up front. This is an inconvenient result of unsatisfactory computer vision systems.

These systems can benefit from understanding the severity of the damage on the exterior of the vehicle more deeply to infer interior damage. This will increase the accuracy of the quantification of the damage and reduce the frequency of insureds being lowballed by insurance companies.

Now, some of the people doing this research are (Fouad et al. 2022), (Parhizkar et al. 2022), (van Ruitenbeek et al. 2022), and (Wang et al. 2023).

(Fouad et al. 2022) investigated pre-trained neural networks VGG-19 and DenseNet-169 for automatic classification of vehicle damages to make insurance claims much faster and more efficient. They proposed a model to improve the feature extraction process. They achieved 81% accuracy with DenseNet-169 and 79% accuracy with VGG-19. When mixing transfer and ensemble learning approaches the final approach has an accuracy of 85.5%.

(Parhizkar et al. 2022) presented an automated system for the recognition of damaged surface parts of cars based on a two-path convolutional neural network. They used a ResNet-50 at the beginning of each route to explore low-level features efficiently and proposed a new mReLU and inception blocks in each route that are responsible for extracting high-level visual features. They compared their proposed method to 8 other methods and outperformed them all in detecting different damaged parts of a car with an accuracy of 93% where accuracy is defined as [(true positive/false positive + true positive) * 100%].

(van Ruitenbeek et al. 2022) compared damage detection of a neural network with damage detection from a human domain expert. They had comparable accuracy with the neural network performing better at detecting the class bend and cover damage. The domain expert was better at detecting the classes dent and scratch. The model used 1 minute on a CPU to process 100 images whereas the domain experts used approximately 2 hours and 15 minutes.

(Wang et al. 2023) created a new publicly available large-scale car damage dataset with extensive annotations for damage detection and segmentation. The lack of a large dataset has been holding back a lot of progress in vehicle damage assessment.

III. Techniques

III.1. Principles, Concepts, and Theoretical Foundations of the research problem

Given an image I of a car, the model assumes that the image contains damage.

The first objective is to locate the exterior damage: $f(I) \rightarrow \{1,2,3,4,5,6\}$ where f is a neural network that maps the image I to a 1 (Front End (front bumper, hood, front grille, front fenders, front lights, front windshield)), 2 (Sides (doors, side mirrors, side fenders, windows, side skirts, pillars)), 3 (Rear end (rear bumper, trunk lid, tailgate, rear lights, rear windshield, rear spoiler)), 4 (Roof (roof panel, sunroof, roof rails)), 5 (Undercarriage (front and rear axles, exhaust system, chassis/frame)), or 6 (Wheels and tires (alloy rims, wheel covers, tires)).

The next step is to classify the exterior damage type: $f(I) \rightarrow \{1,2,3,4,5,6,7,8\}$ where f is a neural network that maps the image I to a 1 (Dent), 2 (Scratch/Scuff/Chip), 3 (Broken Glass), 4 (Headlight Broken), 5 (Taillight Broken), 6 (Smash), 7 (Rust), or 8 (Puncture).

The third step is to classify the exterior damage severity: $f(I) \rightarrow \{0,1,2,3,4\}$ where f is a neural network that maps the image I to a 0 (No Damage), 1 (Minor Damage), 2 (Moderate Damage), 3 (Major Damage), or 4 (Totaled).

The fourth step is locating the interior regions that may be affected: $f(I) \rightarrow \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}$ where f is a neural network that maps the image I to a 1 (Dashboard Area (Instrument panel, center console, globe compartment)), 2 (Front Area (driver\'s seat, passenger seat, gear shift, pedals, steering wheel and column, footwells)), 3 (Middle Area (rear seats, center armrest, floor console)), 4 (Storage and Utility (door pockets, cup holders, rear parcel shelf, boot/trunk)), 5 (Roof Area (sun visors, rearview mirror, sunroof, roof lining, grab handles)), 6 (Rear area (backseat pockets, rear air vents, rear center console)), 7 (Miscellaneous (lighting, ports and outlets, controls)), 8 (Engine compartment (engine, radiator, battery, air filter, fuse box)), 9 (Transmission and Drivetrain (transmission, drive shaft, differential, axles)), 10 (Exhaust system (exhaust manifold, catalytic converter, muffler, exhaust pipe)), 11 (Fuel System (fuel tank, fuel pump, fuel injectors)), 12 (Brake system (brake pedal, master cylinder, brake pads, rotors, brake calipers)), 13 (Suspension and steering (shock absorbers/struts, springs, control arms, steering rack and pinion)), 14 (electrical system (alternator, starter motor, wiring harness)), 15 (climate system (air conditioning compressor, heater core, blower motor)), or 16 (Body and frame (chassis, frame, body panels)).

The fifth step is to classify the interior damage severity: $f(I) \rightarrow \{0,1,2,3,4\}$ where f is a neural network that maps the image I to a 0 (No Damage), 1 (Minor Damage), 2 (Moderate Damage), 3 (Major Damage), or 4 (Totaled).

The last step is deterministically calculating the cost based on predefined mappings.

III.2. Techniques that have been used by other researchers for the research problem

Techniques described by other researchers:

- Damage detection with a two-stage model combined with a weakly supervised segmentation model
    - (Zhang et al. 2020)
- Damage Component Localization to localize damage at the pixel level
    - (Zhang et al. 2020)
    - (Patil et al. 2017)

- Decision and Repair Plan Determination to estimate costs
    - (Zhang et al. 2020)
- Dataset creation
    - (Zhang et al. 2020) Capturing high-quality video with front-end application
    - (Kyu et al. 2020) Web scraping
    - (Wang et al. 2023) Raw data collection from Flickr and Shutterstock
    - (van Ruitenbeek et al. 2022) Dossiers, web scraping, and real-world camera setup
    - (Fouad et al. 2022) Web scraping from Google, Bind, and DuckDuckGo
    - (Patil et al. 2017) Web scraping
- Define Categories
    - (Kyu et al. 2020) Minor, moderate, and severe damage
    - (van Ruitenbeek et al. 2022) bend, bump, cover damage, crack, dent, glass shatter, hail, light broken, missing, rust, scratch, tire crack
    - (Fouad et al. 2022) Broken glass, broken headlights, broken taillights, dents, and scratches
    - (Patil et al. 2017) bumper dent, door dent, glass shatter, headlamp broken, tail lamp broken, scratch, and smash
- Data Augmentation
    - (Kyu et al. 2020) random rotations, zooming, dimensional shifting, and flipping
    - (Parhizkar et al. 2022) geometric transformations, brightness and mirror transformations, Gaussian Noise, random elastic deformations, and random intensity variations
    - (Patil et al. 2017) random rotations, horizontal flip transformations
    - (van Ruitenbeek et al. 2022) remove duplicates, clean data to ensure damage presence, data split 80% training and 20% validation
    - Raap, M. (2021) remove unsuitable images
    - (Zhang et al. 2020) Noise Reduction in video frames
- Pre-trained VGG models
    - (Kyu et al. 2020) VGG16 and VGG19
    - (Parhizkar et al. 2022)
    - (Fouad et al. 2022)
- Convolutional Neural Network

- (Parhizkar et al. 2022)
  - (Fouad et al. 2022)
  - (Patil et al. 2017)
  - Raap, M. (2021)
- Semi-Supervised Mask R-CNN with Saliency propagation modules
  - Raap, M. (2021)
- mReLU model
  - (Parhizkar et al. 2022)
- Transfer Learning
  - (Kyu et al. 2020)
  - (van Ruitenbeek et al. 2022)
  - (Parhizkar et al. 2022)
  - (Fouad et al. 2022)
  - (Patil et al. 2017)
- Ensemble methods
  - (Patil et al. 2017)
- Fine-Tuning
  - (van Ruitenbeek et al. 2022)
- Regularization
  - (Kyu et al. 2020) L1, L2, dropouts, and batch normalization
  - (Fouad et al. 2022) dropouts and batch normalization
  - (Patil et al. 2017) dropouts
- Data Annotation
  - (Wang et al. 2023) 20 person annotation and quality control team
  - (van Ruitenbeek et al. 2022) manual annotation
  - (Patil et al. 2017) manual annotation
  - Raap, M. (2021) COCO annotation for 10% of images
- Optimize Hyperparameters
  - (van Ruitenbeek et al. 2022)
- Comparison
  - (van Ruitenbeek et al. 2022) methods vs methods, models vs models, and models vs experts

- Hypothesis
    - Raap, M. (2021)

Techniques used in this research:

- Hypothesis for what the research aims to prove or disprove
- Dataset creation for collecting a wide variety of car damage images
- Creation of Ontology classifying different types of damage
- Data Annotation labeling images into distinct categories
- Model Creation to detect and classify damage using deep convolutional neural networks
- Regularization for techniques like L1, L2 regularization and dropout to prevent overfitting
- Optimizing Hyperparameters to improve model performance
- Damage Localization determining where the damage is
- Damage Severity determining how severe the damage is
- Repair Cost Estimation using deterministic function based on the assessed damage

The primary aim is to accurately classify and quantify damage. The objective function can be expressed as: $g(exterior\ damage, exterior\ severity, interior\ damage, interior\ severity) \rightarrow Cost$

III.3. Relevant technologies that would be useful to this research

The technologies that were useful to this research would be working knowledge of computer vision, dataset creation, and neural networks. Python and python libraries like keras, tensorflow, and opencv are relevant.

Other techniques that are potentially useful are transfer learning, ensemble techniques, attention mechanisms, generative adversarial networks to generate synthetic data, capsule networks as an alternative to convolutional neural networks, few-shot learning, zero-shot learning, reinforcement learning, meta-learning, and explainable methods.

III.4. Algorithms, Process modules, and solution methods

The neural networks use matrix multiplication in dense layers where $y = \sigma(Wx + b)$ where W is the weight matrix, x is the input vector, b is the bias vector, and σ is the activation function. These neural networks also use convolutional layers where the operation can be represented as $(I * K)(x, y) = \sum_m \sum_n I(x - m, y - n)K(m, n)$ where I is a segment of the input image and K is the filter. The activation functions are ReLU represented as $f(x) = \max(0, x)$ used to introduce non-linearity, sigmoid represented as $S(x) = \frac{1}{1+e^{-x}}$ is the output layer for values for multi-class classification, and softmax represented as $S(x) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$ to perform multi-class classification. MaxPooling layers are also used to reduce spatial dimensions. Flatten layers are used to reshape the data so that it can be processed by a later dense layer. The loss function binary cross-entropy is used to penalize false positives and false negatives represented as $BCE = -ylog(p) - (1 - y)\log(1 - p)$ where p is the predicted value and y is the true value. Categorical Cross-Entropy and sparse categorical cross-entropy are also used where $CE = -t_1 \log(f(s_1)) - (1 - t_1)\log(1 - f(s_1))$. The optimizer used is adam due to its adaptability and efficiency.

IV. Approaches

IV.1. Methodologies I applied in this research

To tackle the problem, I surveyed other approaches to the task. I ended up creating a dataset from scratch with a custom ontology. The dataset was applied to a set of three custom models I created. The code was implemented with Python where the goal was to create a script that used the models to understand the damage to a car in an image and estimate how much repairs costs would be.

IV.2. Techniques I used to solve the problem

Supervised learning with convolutional neural networks is used to detect car damage, localize it, determine its severity, and estimate repair costs. Images of cars with varying types of degrees of damage, makes, models, colors, lighting, shadows, reflections, and weather were collected from publicly available datasets and scraped from the web. The categories of the kind of damage, where it is on the car, and the severity of the damage will be created based on the dataset. The data will be annotated as much as time will allow. Hyperparameters were optimized to increase model performance.

The most novel part of this research is the emphasis on inferring interior damage from exterior damage. The models are simple and roughly based on previous techniques that performed well in the past. Automated repair cost estimation is not new. This is an extension of previous research.

Algorithmic Form of Techniques:

- Define the problem statement
- Propose a preliminary hypothesis regarding car damage detection and repair cost estimation
- Collect images/samples of cars with varying degrees of damage.
- Store and annotate images in labelbox.
- Define categories in a new ontology
- For each image in the dataset
    - Mark the region of damage (if any).
    - Label the type and severity of damage.
- Export data to be used in model training.
- Define neural network architecture or choose an existing architecture.
- Preprocess Data
- Train models
- Select hyperparameters that give the best performance.
- Store models
- For each image in the test dataset
    - Predict what kind of exterior and interior damage it has
    - Predict severity of exterior and interior damage
    - Calculate the cost of repairs
    - Display predictions

IV.3. Processes I engaged in this research

The program will be developed by thinking through the desired result and finding a path to get there. The data had to be manually assembled and annotated to allow for class mapping to the interior of the car. There were no datasets available that tried to infer interior damage from exterior damage so a custom ontology/schema was created to structure the classifications in a useful way. The ontology/schema can be found in the appendix. The data was collected from Google images and Kaggle datasets https://www.kaggle.com/datasets/lplenka/coco-car-damage-detection-dataset/. The data is split into a train dataset to train the models, and a test dataset to evaluate the final performance of the model. The images were uploaded to labelbox where they were manually annotated. An example of how the annotations look in labelbox is in the appendix. The data was then exported for use in training convolutional neural networks. The raw form of the data can be seen in the appendix. The first step of the system is to import and preprocess the data from an .ndjson file. The preprocessed form of the data can be seen in the appendix. I have never used the Labelbox software so custom logic was required to transform the data to a format that the models could work with. Sometimes there were multiple classes of damage present in an image and sometimes there was only one, so the data was padded with zeros to make this easier. The models designed are convolutional neural networks (CNN) with multiple output layers, tailored for a specific use case in predicting car damage from images. The structure of the models can be found in the appendix and reflects their purpose to classify the type and location of the damage. The models were trained in Google Colab using the A100 and V100 GPUs. This is because the models need high levels of GPU RAM to finish training. The training output and loss graphs can be found in the appendix. The models were saved so that less money was spent retraining the models. Test images were used to evaluate if the model could accurately predict what kind of damage was in each image, the severity, and then the cost of all damage combined. The predictions were then graphed with Matplotlib so they were easier to understand and can be found in the appendix. The cost dictionary and function details can be found in the appendix.

The experiments were conducted in a flow. The dataset was prepared. The Model was developed. The model was trained. Visualizations were created from the results. Based on the results, the models, parameters, and data can be refined and improved.

The tests were executed and evaluation metrics were collected. All tests were executed within the Python code or through manual examination.

The experimentation and test results were visualized with Matplotlib, collected, and stored for review.

The criteria to judge whether the prototype is working correctly is based on the metrics and comparing the model predictions to the actual manually annotated answers.

IV.4. Facilities and supplies used for this research

The Python code was written using the following libraries: json, numpy, OpenCV, requests, PIL, io, TensorFlow, Keras, and matplotlib. The images were collected from a google image search. The image annotation was done manually in the Labelbox software. The Python code was written in Google Colab. A premium Google Colab subscription was required so that the models could be trained with an A100 GPU and V100 GPU that had access to a lot of RAM. Google Drive was used to store the models.

V. Workplan

V.1. Tasks performed in this research

Tasks performed:

- Define the problem statement (1 day)
- Propose a preliminary hypothesis regarding car damage detection and repair cost estimation (1 day)
- Collect images/samples of cars with varying degrees of damage (28 days) Expect large dataset of 80+ images
    - Store in an organized directory structure.
- Define categories (1 day)
- Organize dataset based on these categories (1 day)
- For each image in the dataset (28 days) Manual annotation is time consuming

    - Mark the region of damage (if any).
    - Label the type and severity of damage.
- Split dataset into train and validation splits (0 days)
- Define neural network architecture or choose an existing architecture (3 days) Expected a complicated architecture
- Introduce techniques like dropout (1 day)
- Initialize model parameters (0 days)
- Store model performance.
- For each image detected with damage (0 days)
    - Identify and mark the region of damage.
- For each image detected with damage (0 days)
    - Predict the severity of the damage based on defined categories.
- Using the damage type, severity, and localization: (0 days)
    - Estimate the cost of repair using predefined metrics/criteria.
- Compare model predictions with actual labels. (1 day) Expecting to take some time for analysis
- Evaluate performance using metrics like accuracy, precision, recall, etc. (1 day) Expecting to take some time for analysis

I accomplished everything included in this report. I have created and annotated a small dataset of 72 images with damage. The first 64 images are used to train the models with batch size 32. The other 8 images are test data.

I did not accomplish putting together a large dataset (3500+ images) that would allow me to accurately tell if the models are working correctly. Finding and annotating images took much longer than expected. I did not apply various transformations (rotation, flipping, zooming) or store augmented images to increase dataset size. I did not use a validation set of data because I combined it with the training data to make that dataset bigger. I did not have time to add in L1 and L2 regularization and get it working properly. I did not do any freezing of the initial layers for feature extraction. I did not train the models with multiple different hyperparameters because the free version did not have enough memory to complete the training. I did not evaluate the performance using anything else other than accuracy and loss. I did not have time to compare with existing methods or baselines. I did not use any pre-trained models.

V.2. Schedule, timeline, and milestones

Below is a timeline of the schedule for accomplishing the tasks.

Schedule of research conducted

Order	Dates	Task/Activity	Prerequisites (Knowledge, Skill, or Tools)	Expected results
1	Aug 21 – Sept 7	Proposal	Learning about the problem	Have solid plan of execution for the research
2	Sept 8 – Oct 1	Create and Annotate Dataset	Where to find datasets	Dataset of 80+ annotated images
3	Oct 2 – Oct 12	Mid-Term Report	Clear path for research	Have solid plan to finish research
4	Oct 13 – Oct 31	Create Models	Python and libraries	Have working models
5	Nov 1 – Nov 22	Research and Analysis	Data visualization	Improve working models, increase size of dataset
6	Nov 23 – Dec 7	Final Report	Ability to finish research	Demonstrate damage detection and quantification system

VI. Results and Analysis

VI.1. Results

I am using Python code and a manually created dataset to obtain the results. The dataset is made up of many images of cars and all the annotations added to the cars. I am using labelbox to annotate the dataset. The models are created with keras and tensorflow. The output images are plotted with Matplotlib. The results are plotted with the original image with predicted bounding boxes on it, then all the classifications that were predicted to the right of the original image. This can be found in the appendix.

The results of the predictions of the model can be found in the appendix. The output from the models is compared to a manually annotated version of the same image. The box labels are all unknown and the bounding boxes highlight essentially the entire image. This prediction is not accurate. The exterior location labels are all 'Unknown'. This is not accurate. There may be a problem with the data or model for this, otherwise, more data should help. The exterior damage severity and interior damage severity seem like it is pretty accurate. It is coming up with predictions that are often correct for the test data. The interior damage location labels are always 'climate system' which is not accurate. The cost is working as expected but is off by more than expected because the cost of an 'Unknown' is 0.

The performance metrics used are loss, accuracy, and mean squared error. The performance was also visually evaluated using the output images. The values for these can be found in the model training section of the appendix.

VI.2. Analysis of the results

A quantitative analysis can be obtained using metrics loss, accuracy, and mean squared error. The Exterior damage location and damage types model did not seem to train correctly. This could be a result of the imbalanced dataset, issues with the format of the data, or issues with the model outputs. The other two models trained reasonably well. The loss curves can give insight into possible overfitting.

Damage Location for both exterior and interior as well as classifying damage types suffered greatly from lack of data. More data would have helped better identify where the model can benefit from improvement. Overall, the project performed poorly and I would not consider this research successful.

The results are significant because they show a proof of concept for the idea of inferring interior damage from images of exterior damage. It illustrates the challenges with a task like this and some of the current limitations. It contributes to advancing the understanding of inference with images of vehicles. There is a clear path to improve upon this for future work.

Compared to other researchers' work, like (Patil et al. 2017) and (Kyu et al. 2020) this research performed extremely poorly. I was not able to achieve anywhere close to the accuracy achieved by these other researchers. However, the scope of this research in comparison was much broader. (Patil et al. 2017) was only doing damage classification on the car exterior which was 1 of 6 pieces of this research. (Kyu et al. 2020) did damage detection, damage localization, and damage severity on the exterior only which is 3 of 6 pieces of this research. In hindsight, it would have been better to have tried to extend the research of (Kyu et al. 2020) rather than build everything from scratch.

VI.3. Discussions

VI.3.1. Problems and issues arisen during the research

Dataset creation has been difficult for many previous researchers and has been difficult in this research. I created all the annotations to get the results I am looking for manually which is the most time-consuming part of this research. It is likely the model overfit and will only work on the dataset provided. I do not think this issue can be mitigated by using data augmentation techniques and regularization techniques until the initial dataset is larger to begin with. I calculated that with 40 classes in my dataset that are needed for defining the damage present in an image, an annotation speed of 37 images per hour, using 6 classes per image, and 500 images per class group, I would need 3500 images, which would take 95 hours to mark them all up. This is not feasible in the timeframe of this research. The model results should not be expected to be accurate with this little training data which makes them hard to interpret and makes it hard to improve the models.

I do not have access to a GPU on my personal machine which caused computational constraints. I did not have any data on the cost of car based on exterior damages so the current costs are estimates based on loose ranges.

To solve some of these problems, I have decided to stop dataset creation at 72 images so that I have enough time to get the models working and write the final report. I purchased access to GPUs through google colab. The targets of the research have evolved to incorporate these issues.

VI.3.2. Limitation and constraints of the research (and results)

I am limited by the amount of time available to work on this project since I am working full-time. I am limited in my knowledge and skills since I am not a computer vision expert or AI expert or Python expert. There may be limitations in the generalizability of the model. Over time cars change appearance and the models will likely lose performance over time unless retrained regularly on a new dataset. I am limited by the tools I must perform the research. The results may have limited utility to a practitioner.

The model is designed on still images and will not be suitable for video feeds without adaptation. The system will work best in a controlled environment with specific lighting and standardized image capture settings. If the data is only of cars in a certain region, it will not perform well when applied in a different region.

VII. Summary

Car accidents often result in exterior damage, which is observable. However, the exterior damage often conceals deeper, internal damages that are not immediately visible. There is a need for an automated solution that can, by analyzing the exterior damage, predict potential internal damages and estimate the repair costs accurately. This would speed up the insurance claim process and help repair shops in initial assessments.

Currently, damage assessments are typically conducted manually by professionals who inspect the damaged car and provide an estimate. Automated solutions leveraging computer vision, primarily focused on exterior damage detection, are also in use in some places. However, the inference of interior damages from exterior ones remains a significant challenge. Deep learning techniques have shown promise in related domains but are yet to be fully explored for this specific problem.

Techniques Used:

- Hypothesis for what the research aims to prove or disprove
- Dataset creation for collecting a wide variety of car damage images
- Creation of Categories classifying different types of damage
- Data Annotation labeling images into distinct categories
- Model Creation to detect and classify damage using deep convolutional neural networks
- Use of dropout layers to prevent overfitting
- Damage Localization determining where the damage is
- Damage Severity determining how severe the damage is
- Hierarchical Damage Representation for classifying damage in a layered manner, from exterior to possible linked interior damage.
- Repair Cost Estimation using deterministic estimate

The research produced an automated system that can detect and categorize external damages with an accuracy rate not exceeding current solutions. With improvements to the dataset, the automated system has the potential to provide predictive insights into internal damage based on exterior assessments. Finally, the model results do not contain accurate repair cost estimations because of the lack of data containing cost estimations between exterior and interior damage regions.

The innovation in this research is the novel approach to infer interior damages from observable exterior damages, using deep learning. The research produced a cost estimation model on a small dataset of damages and estimated repair costs.

The expected new contributions to the knowledge and advancement in the related scientific, technological, and engineering areas include an approach to inference in the domain of computer vision. This research attempted to bridge the gap between observable external damages and hidden internal problems, making the car repair process more transparent, swift, and efficient for all stakeholders involved.

Future work could be done on standardizing dataset creation, enlarging the dataset, attempting to optimize the models and workflow presented, developing more interpretable results, making the model robust against adversarial attacks or unusual inputs, finding ways to learn with fewer examples, implementing video feeds, implementing real-time processing, broadening the damage contextualization, and including data from all regions.

References

Fouad, M. M., Malawany, K., Osman, A. G., Amer, H. M., Abdulkhalek, A. M., & Eldin, A. B. (2022). Automated vehicle inspection model using a deep learning approach. *Journal of Ambient Intelligence and Humanized Computing*, 1-9.

Kyu, P. M., & Woraratpanya, K. (2020, July). Car damage detection and classification. In *Proceedings of the 11th international conference on advances in information technology* (pp. 1-6).

Parhizkar, M., & Amirfakhrian, M. (2022). Recognizing the Damaged Surface Parts of Cars in the Real Scene Using a Deep Learning Framework. *Mathematical Problems in Engineering*, *2022*.

Patil, K., Kulkarni, M., Sriraman, A., & Karande, S. (2017, December). Deep learning based car damage classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)* (pp. 50-54). IEEE.

Raap, M. (2021). *Vehicle Damage Detection using Semi-Supervised Object Detection* (Doctoral dissertation, Dublin, National College of Ireland).

van Ruitenbeek, R. E., & Bhulai, S. (2022). Convolutional Neural Networks for vehicle damage detection. *Machine Learning with Applications*, *9*, 100332.

Wang, X., Li, W., & Wu, Z. (2023). CarDD: A New Dataset for Vision-Based Car Damage Detection. *IEEE Transactions on Intelligent Transportation Systems*.

Zhang, W., Cheng, Y., Guo, X., Guo, Q., Wang, J., Wang, Q., ... & Chu, W. (2020, April). Automatic car damage assessment system: Reading and understanding videos as professional insurance inspectors. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 09, pp. 13646-13647).

Appendix

Dataset

One record of the raw dataset looks like the following:


```
{'data_row': {'id': 'clna8douq03km0784wcgnz6n3',
  'external_id': '2.jpg',
  'row_data':
'https://storage.labelbox.com/clna88c400kn207246g12a2y5%2F5d71ef10-b8a7-
cf7e-f6d6-af8bb0cfb581-2.jpg?Expires=1700500575237&KeyName=labelbox-
assets-key-3&Signature=3pIhLaYJHEHq7nbMotMZKZNm7Pw',
  'details': {'dataset_id': 'clna8dhkv000o0704wuea2jn6',
   'dataset_name': 'Train',
   'created_at': '2023-10-03T11:22:57.391+00:00',
   'updated_at': '2023-10-03T11:22:59.042+00:00',
   'last_activity_at': '2023-10-04T23:42:45.296+00:00',
   'created_by': 'austinvornhagen@unomaha.edu'}},
 'media_attributes': {'height': 1024,
  'width': 1024,
  'mime_type': 'image/jpeg',
  'exif_rotation': '1'},
 'attachments': [],
 'metadata_fields': [],
 'projects': {'clna8cr1l0i0107z19feu8tpt': {'name': 'Car Damage
Detection',
   'labels': [{'label_kind': 'Default',
     'version': '1.0.0',
```

'id': 'clna9nrz40f7c07yv79r8f2ie',

'label_details': {'created_at': '2023-10-04T10:11:07.000+00:00',

'updated_at': '2023-10-04T23:41:12.000+00:00',

'created_by': 'austinvornhagen@unomaha.edu',

'content_last_updated_at': '2023-10-04T23:41:07.454+00:00',

'reviews': []},

'performance_details': {'seconds_to_create': 86,

'seconds_to_review': 17,

'skipped': False},

'annotations': {'objects': [{'feature_id':
'clnbl7zg6000n356ovwyigysn',

'feature_schema_id': 'clna9mw4328sa070nbtbs8k6b',

'name': 'Dent',

'annotation_kind': 'ImageBoundingBox',

'classifications': [],

'bounding_box': {'top': 500.0,

'left': 612.0,

'height': 254.0,

'width': 279.0}},

{'feature_id': 'clnbl87ww000q356om64pxbli',

'feature_schema_id': 'clna9mw4328sc070nhb008mkq',

'name': 'Scratch/Scuff/Chip',

'annotation_kind': 'ImageBoundingBox',

'classifications': [],

'bounding_box': {'top': 861.0,

```
          'left': 652.0,
          'height': 42.0,
          'width': 91.0}},
        {'feature_id': 'clnce6slb0002356oo3uk112m',
         'feature_schema_id': 'clna9mw4328si070n9w5zap16',
         'name': 'Taillight broken',
         'annotation_kind': 'ImageBoundingBox',
         'classifications': [],
         'bounding_box': {'top': 138.0,
          'left': 547.0,
          'height': 198.0,
          'width': 249.0}}],
       'classifications': [{'feature_id': 'clnbl8eg4000t356ocsqbww4m',
         'feature_schema_id': 'clna9mw4328sm070nb5zfd0ky',
         'name': 'Damage',
         'radio_answer': {'feature_id': 'clnbl8eg4000s356okmcdwz4t',
          'feature_schema_id': 'clna9mw4328sn070n0ik36nc1',
          'name': 'Yes Damage',
          'classifications': []}},
        {'feature_id': 'clnbl8k6c000v356od6hf2r6b',
         'feature_schema_id': 'clna9mw4428ss070nctpoa5sj',
         'name': 'Exterior Location',
         'checklist_answers': [{'feature_id': 'clnbl8k6c000u356okwvp8div',
           'name': 'Rear end (rear bumper, trunk lid, tailgate, rear
lights, rear windshield, rear spoiler)',
```

'classifications': []}]},
{'feature_id': 'clnbl8p15000x356o6xnkmr3w',
'feature_schema_id': 'clna9mw4428t6070n73u3a03m',
'name': 'Exterior Damage Severity',
'radio_answer': {'feature_id': 'clnbl8p15000w356o7di03877',
'feature_schema_id': 'clna9mw4428t9070n8wyl7ps1',
'name': 'Minor Damage',
'classifications': []}},
{'feature_id': 'clnbl8wtg000z356o0ekgijpj',
'feature_schema_id': 'clna9mw4428ti070n9t3x34pg',
'name': 'Potential Interior Regions Affected',
'checklist_answers': [{'feature_id': 'clnbl8wtg000y356od9sbr7wu',
'name': 'Storage and Utility (door pockets, cup holders, rear parcel shelf, boot/trunk)',
'classifications': []},
{'feature_id': 'clnbl8yhc0011356owcwv30qs',
'name': 'Body and frame (chasis, frame, body panels)',
'classifications': []}]},
{'feature_id': 'clnbl928y0013356oilbl22wr',
'feature_schema_id': 'clna9mw4528ug070n8xhihdo7',
'name': 'Interior Damage Severity',
'radio_answer': {'feature_id': 'clnbl928y0012356og5aql0yx',
'feature_schema_id': 'clna9mw4528uh070nfofe0of5',
'name': 'No damage',
'classifications': []}}],

```
    'relationships': []}}],
  'project_details': {'ontology_id': 'clna8l7p429hh07zghjcqaqfv',
   'task_name': 'Done',
   'batch_id': 'a28c0270-61df-11ee-9f27-2d11f76203e6',
   'batch_name': 'car_damage_detection_batch_1',
   'workflow_status': 'DONE',
   'priority': 5,
   'consensus_expected_label_count': 1,
   'workflow_history': [{'action': 'Approve',
     'created_at': '2023-10-04T23:42:45.308+00:00',
     'created_by': 'austinvornhagen@unomaha.edu',
     'previous_task_name': 'Initial review task',
     'previous_task_id': 'a49ef1fa-2b48-4089-b7e2-9ddf94bf8ab2'},
    {'action': 'Rework',
     'created_at': '2023-10-04T23:41:12.876+00:00',
     'created_by': 'austinvornhagen@unomaha.edu',
     'previous_task_name': 'Rework (all rejected)',
     'previous_task_id': '5fee4303-e240-0dd7-b0ed-e6edc6896e46',
     'next_task_name': 'Initial review task',
     'next_task_id': 'a49ef1fa-2b48-4089-b7e2-9ddf94bf8ab2'},
    {'action': 'Reject',
     'created_at': '2023-10-04T23:34:41.246+00:00',
     'created_by': 'austinvornhagen@unomaha.edu',
     'previous_task_name': 'Initial review task',
     'previous_task_id': 'a49ef1fa-2b48-4089-b7e2-9ddf94bf8ab2',
```

```
       'next_task_name': 'Rework (all rejected)',
       'next_task_id': '5fee4303-e240-0dd7-b0ed-e6edc6896e46'},
      {'action': 'Move',
       'created_at': '2023-10-04T10:11:07.142+00:00',
       'created_by': 'austinvornhagen@unomaha.edu',
       'previous_task_name': 'Initial labeling task',
       'previous_task_id': '21fdeb9a-68cc-04c2-afca-4b003035b97a',
       'next_task_name': 'Initial review task',
       'next_task_id': 'a49ef1fa-2b48-4089-b7e2-9ddf94bf8ab2'},
      {'action': 'Move',
       'created_at': '2023-10-04T10:11:07.126+00:00',
       'created_by': 'austinvornhagen@unomaha.edu',
       'next_task_name': 'Initial labeling task',
       'next_task_id': '21fdeb9a-68cc-04c2-afca-4b003035b97a'}]}}}}
```

The preprocessing step removes the excess data that is not useful in training the models for damage classification. The only data that is pulled is the image filename, the image pixels from the url, the image height, image width, the original dataset name, the annotation objects, and the classifications. One record from the processed dataset looks like:

```
{'image_file': '2.jpg',
 'image_pixels': [(253, 252, 250),
  (253, 252, 250),
  (253, 252, 250),
  (253, 252, 250),
  (253, 252, 250),
  (253, 252, 250),
```

```
  (253, 252, 250),
  (253, 252, 250),
  (254, 253, 251),
  (254, 253, 251),
  (254, 253, 251),
  (254, 253, 251),
  ...],
 'image_height': 1024,
 'image_width': 1024,
 'original_dataset': 'Train',
 'objects': [('Dent',
   {'top': 500.0, 'left': 612.0, 'height': 254.0, 'width': 279.0}),
  ('Scratch/Scuff/Chip',
   {'top': 861.0, 'left': 652.0, 'height': 42.0, 'width': 91.0}),
  ('Taillight broken',
   {'top': 138.0, 'left': 547.0, 'height': 198.0, 'width': 249.0})],
 'classifications': [('Damage', 'Yes Damage'),
  ('Exterior Location',
   'Rear end (rear bumper, trunk lid, tailgate, rear lights, rear windshield, rear spoiler)'),
  ('Exterior Damage Severity', 'Minor Damage'),
  ('Potential Interior Regions Affected',
   'Storage and Utility (door pockets, cup holders, rear parcel shelf, boot/trunk)'),
  ('Potential Interior Regions Affected',
   'Body and frame (chasis, frame, body panels)'),
```

```
('Interior Damage Severity', 'No damage')]}
```

The dataset was not balanced. Here are tables of the distribution of the dataset:

Damage Type	**Count**	**% Share**
Scratch/Scuff/Chip	124	38.99%
Dent	99	31.13%
Smash	39	12.26%
Headlight Broken	17	5.35%
Rust	13	4.09%
Puncture	11	3.46%
Broken Glass	8	2.52%
Taillight Broken	7	2.20%
Total	**318**	**100%**

Exterior Location	**Count**	**% Share**
Front End (…)	34	37.78%
Rear End (…)	26	28.89%
Sides (…)	25	27.78%
Wheels and Tires (…)	4	4.44%
Roof (…)	1	1.11%
Undercarriage (…)	0	0.00%
Total	**90**	**100%**

Exterior Severity	**Count**	**% Share**
Minor Damage	45	60.00%
No Damage	0	0.00%
Moderate Damage	27	36.00%
Major Damage	3	4.00%
Totaled	0	0.00%

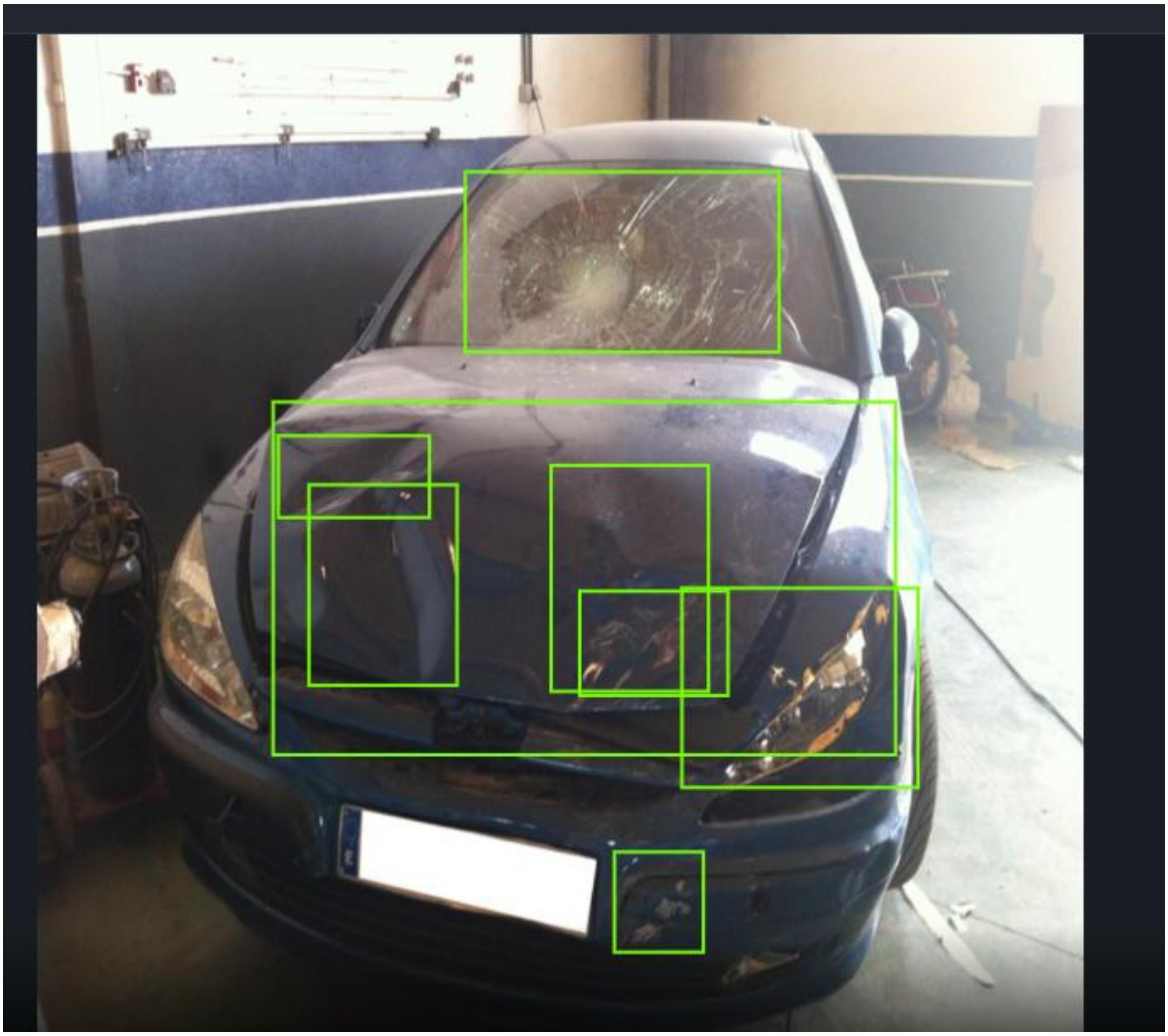
Total	**75**	**100%**

Interior Location	**Count**	**% Share**
Body and Frame (…)	75	43.60%
Engine Compartment (…)	24	13.95%
Storage and Utility (…)	22	12.79%
Electrical System (…)	13	7.56%
Climate System (…)	12	6.98%
Suspension and Steering (…)	8	4.65%
Brake System (…)	6	3.49%
Miscellaneous (…)	4	2.33%
Front Area (…)	3	1.74%
Roof Area (…)	2	1.16%
Exhaust System (…)	2	1.16%
Fuel System (…)	1	0.58%
Dashboard Area (…)	0	0.00%
Middle Area (…)	0	0.00%
Rear Area (…)	0	0.00%
Transmission and Drivetrain (…)	0	0.00%
Total	**172**	**100%**

Interior Severity	**Count**	**% Share**
No Damage	37	49.33%
Minor Damage	28	37.33%
Moderate Damage	9	12.00%
Major Damage	1	1.33%
Totaled	0	0.00%
Total		**100%**

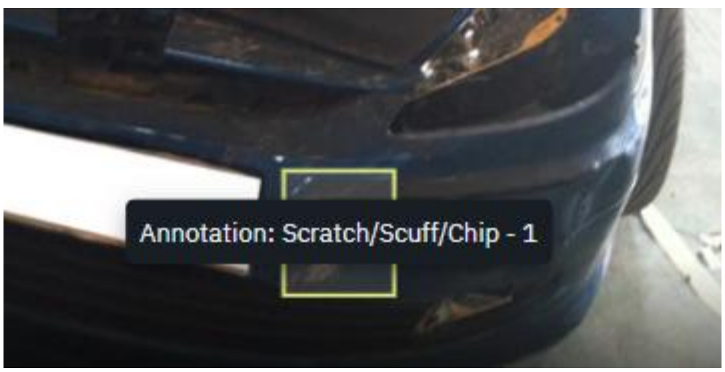
Annotation

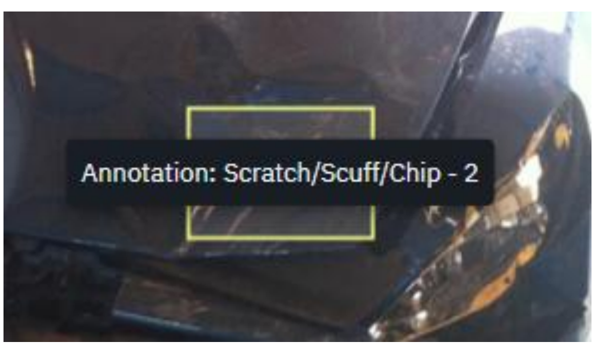
A finished annotation looks like this:



Each object looks like this:


Annotation: Smash


Annotation: Scratch/Scuff/Chip - 1


Annotation: Scratch/Scuff/Chip - 2


Annotation: Headlight broken


Annotation: Dent - 1


Annotation: Dent - 2


Annotation: Dent - 3

Here are the classifications (Moderate… mean 'Moderate Damage'):

Interior Damage Severity : Moderate...

Potential Interior Regions Affected

4 Options

- Body and frame (chasis, frame, body panels)
- climate system (air conditioning compressor, heater core, blower motor)
- electrical system (alternator, starter motor, wiring harness)
- Engine compartment (engine, radiator, battery, air filter, fuse box)

Exterior Damage Severity : Moderate...

Exterior Location

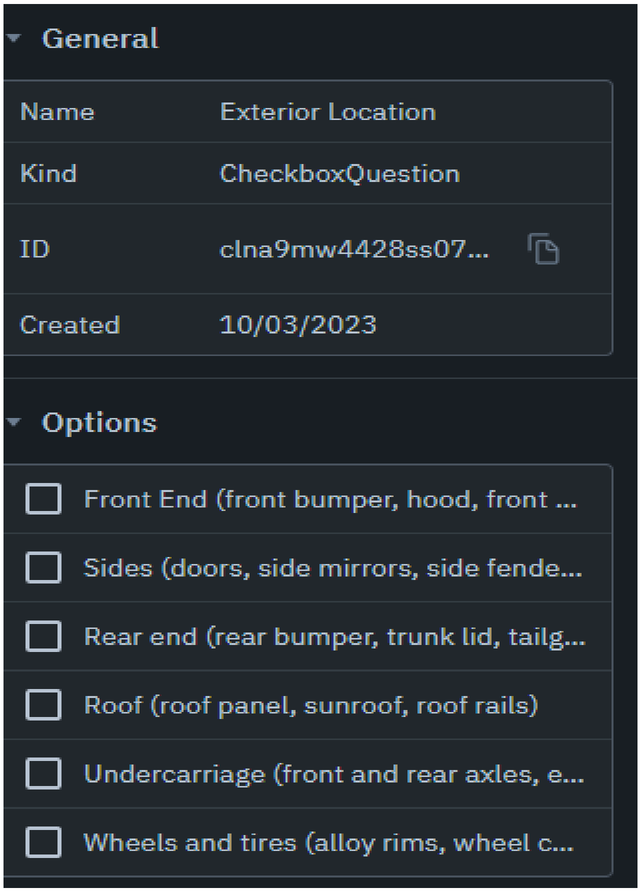
1 Option

- Front End (front bumper, hood, front grille, front fenders, front lights, front windshield)

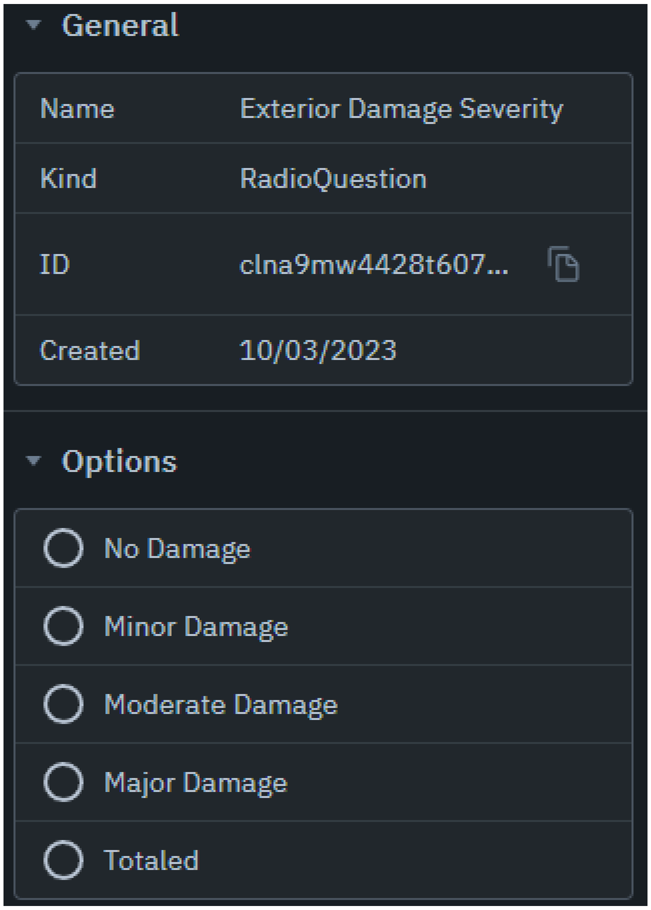
Ontology/Schema

The first part of the Ontology is the objects that are manually drawn on the images. In the training dataset there was at most 11 objects in a single image. There should not be any limit on how many of these are present in an image and any of them can be repeated as many times as necessary:

The next part is the Exterior Damage Location which has 6 options. It is possible to have 1 up to all 6 for this classification since a car can be damaged in all the areas at the same time:



The Exterior Damage Severity classification has 5 options, but only 1 can be chosen. There may be major damage in some places and minor damage is others. This means the choice for this classification is the max damage severity of all the damage present:


General
Name
Exterior Damage Severity
Kind
RadioQuestion
ID
clna9mw4428t607...
Created
10/03/2023
Options
No Damage
Minor Damage
Moderate Damage
Major Damage
Totaled

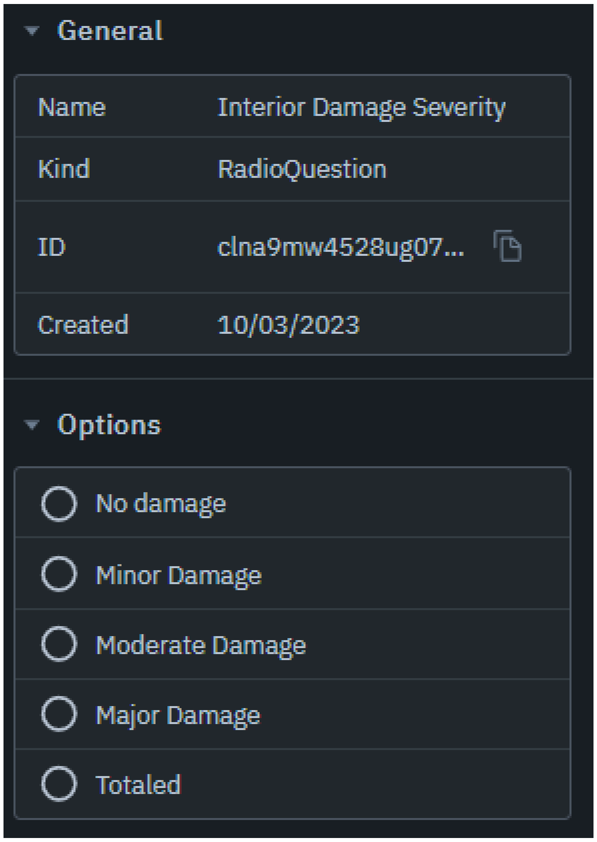
The Interior Damage Location has 16 options to allow for more detailed inference from exterior to interior damage. It is possible to have 1 up to all 16 for this classification since a car can be damaged in all the areas at the same time:

Name	Potential Interior Regio...
Kind	CheckboxQuestion
ID	clna9mw4428ti...
Created	10/03/2023

Options

- [ ] Dashboard Area (Instrument pane...
- [ ] Front Area (driver's seat, passeng...
- [ ] Middle Area (rear seats, center ar...
- [ ] Storage and Utility (door pockets, ...
- [ ] Roof Area (sun visors, rearview mi...
- [ ] Rear area (backseat pockets, rear ...
- [ ] Miscellaneous (lighting, ports and ...
- [ ] Engine compartment (engine, radi...
- [ ] Transmission and Drivetrain (trans...
- [ ] Exhaust system (exhaust manifold...
- [ ] Fuel System (fuel tank, fuel pump,...
- [ ] Brake system (brake pedal, maste...
- [ ] Suspension and steering (shock a...
- [ ] electrical system (alternator, start...
- [ ] climate system (air conditioning c...
- [ ] Body and frame (chasis, frame, bo...

The Interior Damage Severity classification has 5 options, but only 1 can be chosen. There may be major damage in some places and minor damage is others. This means the choice for this classification is the max damage severity of all the damage present:



Model Design

The first model is for Exterior Damage Classification and Location:

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 1024, 1024, 3)]      0         []

 conv2d (Conv2D)             (None, 1022, 1022, 32)       896       ['input_1[0][0]']

 max_pooling2d (MaxPooling2  (None, 511, 511, 32)         0         ['conv2d[0][0]']
 D)

 dropout (Dropout)           (None, 511, 511, 32)         0         ['max_pooling2d[0][0]']

 conv2d_1 (Conv2D)           (None, 509, 509, 64)         18496     ['dropout[0][0]']

 max_pooling2d_1 (MaxPoolin  (None, 254, 254, 64)         0         ['conv2d_1[0][0]']
 g2D)

 dropout_1 (Dropout)         (None, 254, 254, 64)         0         ['max_pooling2d_1[0][0]']

 conv2d_2 (Conv2D)           (None, 252, 252, 128)        73856     ['dropout_1[0][0]']

 max_pooling2d_2 (MaxPoolin  (None, 126, 126, 128)        0         ['conv2d_2[0][0]']
 g2D)

 dropout_2 (Dropout)         (None, 126, 126, 128)        0         ['max_pooling2d_2[0][0]']

 flatten (Flatten)           (None, 2032128)              0         ['dropout_2[0][0]']

 dense (Dense)               (None, 128)                  2601125   ['flatten[0][0]']
                                                          12

 dropout_3 (Dropout)         (None, 128)                  0         ['dense[0][0]']

 box_classes (Dense)         (None, 11)                   1419      ['dropout_3[0][0]']

 ext_classes (Dense)         (None, 3)                    387       ['dropout_3[0][0]']

 boxes (Dense)               (None, 44)                   5676      ['dropout_3[0][0]']

==================================================================================================
Total params: 260213242 (992.63 MB)
Trainable params: 260213242 (992.63 MB)
Non-trainable params: 0 (0.00 Byte)
__________________________________________________________________________________________________
```

The input layer is (1024x1024x3) to process RGB images that are higher resolution. The convolution layers are standard for image recognition tasks. They start at 32 filters to capture low-level features such as edges, corners, and simple textures. Deeper in the network the layers start to combine low-level features to form more complex, high-level features. To detect more complex and varied features, a larger number of filters is necessary so the 64 filters and 128 filters layers were added. The max pooling layers reduce the width and height of the input volume for the next convolutional layer which helps with reducing the computational load. The dropout layers are a regularization technique to reduce overfitting. The flatten layer is used to flatten the output of the convolutional layers from a multi-dimensional tensor into a one-dimensional tensor so it can connect to dense layers. The dense layers are used to perform the classification. There are multiple output layers, one for classifying the type of damage in a box, one for classifying exterior damage location, and one for predicting bounding boxes for the image output to visually show where the damage was detected. There are many parameters so that it can perform complex image classification tasks.

The second model is for Exterior Damage Severity:

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 1024, 1024, 3)]   0

 conv2d_3 (Conv2D)           (None, 1022, 1022, 32)    896

 max_pooling2d_3 (MaxPoolin  (None, 511, 511, 32)      0
 g2D)

 conv2d_4 (Conv2D)           (None, 509, 509, 64)      18496

 max_pooling2d_4 (MaxPoolin  (None, 254, 254, 64)      0
 g2D)

 conv2d_5 (Conv2D)           (None, 252, 252, 128)     73856

 max_pooling2d_5 (MaxPoolin  (None, 126, 126, 128)     0
 g2D)

 flatten_1 (Flatten)         (None, 2032128)           0

 dense_1 (Dense)             (None, 128)               260112512

 dropout_4 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 5)                 645

=================================================================
Total params: 260206405 (992.61 MB)
Trainable params: 260206405 (992.61 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

The layers of this model are similar to the previous model. There is one output layer for classifying the severity of exterior car damage into 5 distinct categories. There are many parameters so that it can perform complex image classification tasks.

The third model is for Interior Damage Location and Severity:

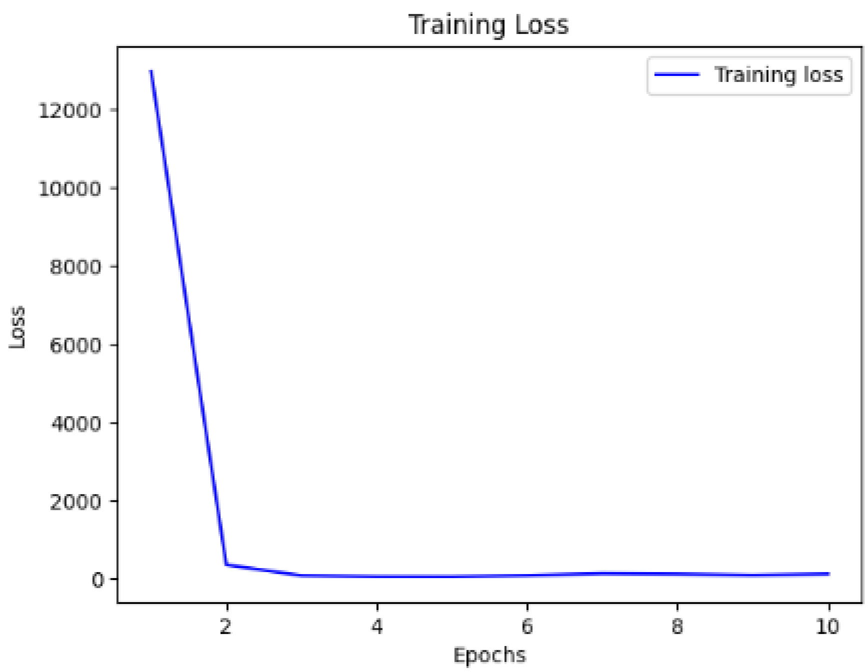
```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_3 (InputLayer)        [(None, 1024, 1024, 3)]      0         []

 conv2d_6 (Conv2D)           (None, 1022, 1022, 32)       896       ['input_3[0][0]']

 max_pooling2d_6 (MaxPoolin  (None, 511, 511, 32)         0         ['conv2d_6[0][0]']
 g2D)

 conv2d_7 (Conv2D)           (None, 509, 509, 64)         18496     ['max_pooling2d_6[0][0]']

 max_pooling2d_7 (MaxPoolin  (None, 254, 254, 64)         0         ['conv2d_7[0][0]']
 g2D)

 conv2d_8 (Conv2D)           (None, 252, 252, 128)        73856     ['max_pooling2d_7[0][0]']

 max_pooling2d_8 (MaxPoolin  (None, 126, 126, 128)        0         ['conv2d_8[0][0]']
 g2D)

 flatten_2 (Flatten)         (None, 2032128)              0         ['max_pooling2d_8[0][0]']

 dense_3 (Dense)             (None, 128)                  2601125   ['flatten_2[0][0]']
                                                          12

 sev_classes (Dense)         (None, 5)                    645       ['dense_3[0][0]']

 int_classes (Dense)         (None, 16)                   2064      ['dense_3[0][0]']

==================================================================================================
Total params: 260208469 (992.62 MB)
Trainable params: 260208469 (992.62 MB)
Non-trainable params: 0 (0.00 Byte)
__________________________________________________________________________________________________
```

The layers of this model are similar to the previous models. There are multiple output layers for classifying the severity of interior car damage into 5 distinct categories and classifying the interior damage location into 16 categories. There are many parameters so that it can perform complex image classification tasks.

Model Training

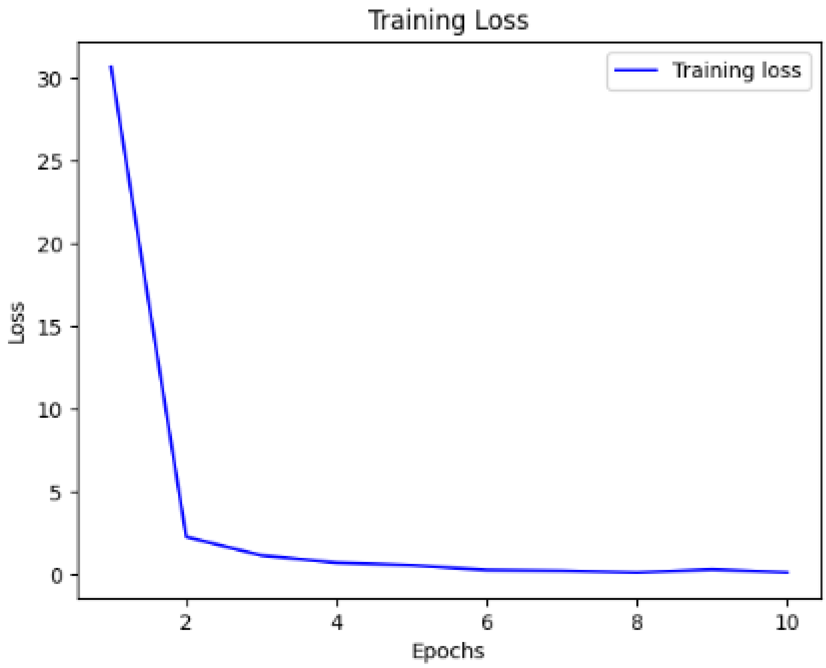
Training for the Exterior Damage Classification and Location Model:

```
Epoch 1/10
4/4 [==============================] - 17s 428ms/step - loss: 12973.4238 - box_classes_loss: 1573.0381 - ext_classes_loss: 158.3791 - boxes_loss: 11242.0068 - box_classes_accuracy: 0.1719 - ext_classes_accuracy: 0.4375 - boxes_mean_squared_error: 11242.0068
Epoch 2/10
4/4 [==============================] - 2s 419ms/step - loss: 346.3702 - box_classes_loss: 165.4044 - ext_classes_loss: 20.8648 - boxes_loss: 160.1009 - box_classes_accuracy: 0.1562 - ext_classes_accuracy: 0.3750 - boxes_mean_squared_error: 160.1009
Epoch 3/10
4/4 [==============================] - 2s 416ms/step - loss: 70.6537 - box_classes_loss: 48.2850 - ext_classes_loss: 7.7374 - boxes_loss: 14.6313 - box_classes_accuracy: 0.0625 - ext_classes_accuracy: 0.4688 - boxes_mean_squared_error: 14.6313
Epoch 4/10
4/4 [==============================] - 2s 417ms/step - loss: 44.3252 - box_classes_loss: 35.2936 - ext_classes_loss: 3.9851 - boxes_loss: 5.0465 - box_classes_accuracy: 0.0469 - ext_classes_accuracy: 0.5938 - boxes_mean_squared_error: 5.0465
Epoch 5/10
4/4 [==============================] - 2s 413ms/step - loss: 41.9612 - box_classes_loss: 33.5763 - ext_classes_loss: 3.9098 - boxes_loss: 4.4751 - box_classes_accuracy: 0.0625 - ext_classes_accuracy: 0.5781 - boxes_mean_squared_error: 4.4751
Epoch 6/10
4/4 [==============================] - 2s 412ms/step - loss: 73.2449 - box_classes_loss: 53.3027 - ext_classes_loss: 8.7788 - boxes_loss: 11.1634 - box_classes_accuracy: 0.0469 - ext_classes_accuracy: 0.6875 - boxes_mean_squared_error: 11.1634
Epoch 7/10
4/4 [==============================] - 2s 413ms/step - loss: 124.1942 - box_classes_loss: 75.1885 - ext_classes_loss: 9.3820 - boxes_loss: 39.6237 - box_classes_accuracy: 0.1094 - ext_classes_accuracy: 0.7344 - boxes_mean_squared_error: 39.6237
Epoch 8/10
4/4 [==============================] - 2s 413ms/step - loss: 110.2846 - box_classes_loss: 66.5974 - ext_classes_loss: 10.4795 - boxes_loss: 33.2077 - box_classes_accuracy: 0.0469 - ext_classes_accuracy: 0.5781 - boxes_mean_squared_error: 33.2077
Epoch 9/10
4/4 [==============================] - 2s 412ms/step - loss: 80.6184 - box_classes_loss: 52.0236 - ext_classes_loss: 8.1116 - boxes_loss: 20.4832 - box_classes_accuracy: 0.0469 - ext_classes_accuracy: 0.6406 - boxes_mean_squared_error: 20.4832
Epoch 10/10
4/4 [==============================] - 2s 413ms/step - loss: 112.1712 - box_classes_loss: 70.3313 - ext_classes_loss: 7.8726 - boxes_loss: 33.9673 - box_classes_accuracy: 0.0312 - ext_classes_accuracy: 0.5781 - boxes_mean_squared_error: 33.9673
```



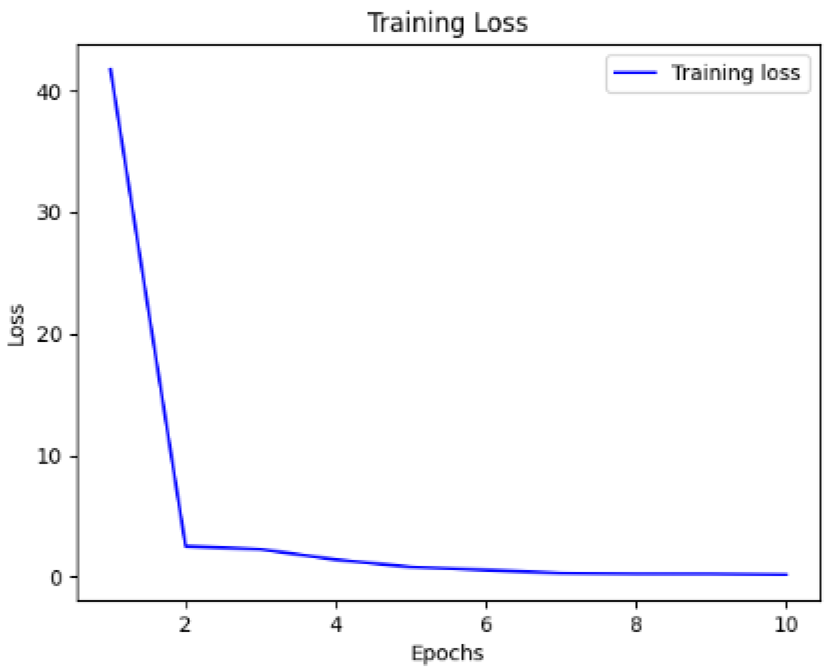
Training for the Exterior Damage Severity Model:

```
Epoch 1/10
4/4 [==============================] - 14s 253ms/step - loss: 30.6713 - accuracy: 0.3594
Epoch 2/10
4/4 [==============================] - 1s 252ms/step - loss: 2.2808 - accuracy: 0.6406
Epoch 3/10
4/4 [==============================] - 1s 252ms/step - loss: 1.1580 - accuracy: 0.5938
Epoch 4/10
4/4 [==============================] - 1s 252ms/step - loss: 0.7163 - accuracy: 0.6562
Epoch 5/10
4/4 [==============================] - 1s 251ms/step - loss: 0.5637 - accuracy: 0.7812
Epoch 6/10
4/4 [==============================] - 1s 254ms/step - loss: 0.2721 - accuracy: 0.8906
Epoch 7/10
4/4 [==============================] - 1s 253ms/step - loss: 0.2281 - accuracy: 0.8906
Epoch 8/10
4/4 [==============================] - 1s 252ms/step - loss: 0.1210 - accuracy: 0.9844
Epoch 9/10
4/4 [==============================] - 1s 252ms/step - loss: 0.3056 - accuracy: 0.9062
Epoch 10/10
4/4 [==============================] - 1s 251ms/step - loss: 0.1290 - accuracy: 0.9688
```



Training for the Interior Damage Location and Damage Severity Model:

```
Epoch 1/10
4/4 [==============================] - 15s 261ms/step - loss: 41.7252 - sev_classes_loss: 37.9880 - int_classes_loss: 3.7372 - sev_classes_accuracy: 0.3281 - int_classes_accuracy: 0.2812
Epoch 2/10
4/4 [==============================] - 1s 257ms/step - loss: 2.5194 - sev_classes_loss: 2.1090 - int_classes_loss: 0.4104 - sev_classes_accuracy: 0.5469 - int_classes_accuracy: 0.3906
Epoch 3/10
4/4 [==============================] - 1s 258ms/step - loss: 2.2720 - sev_classes_loss: 1.8598 - int_classes_loss: 0.4122 - sev_classes_accuracy: 0.5156 - int_classes_accuracy: 0.3906
Epoch 4/10
4/4 [==============================] - 1s 259ms/step - loss: 1.4178 - sev_classes_loss: 0.9750 - int_classes_loss: 0.4428 - sev_classes_accuracy: 0.6094 - int_classes_accuracy: 0.3906
Epoch 5/10
4/4 [==============================] - 1s 259ms/step - loss: 0.8127 - sev_classes_loss: 0.5011 - int_classes_loss: 0.3116 - sev_classes_accuracy: 0.9219 - int_classes_accuracy: 0.3906
Epoch 6/10
4/4 [==============================] - 1s 259ms/step - loss: 0.5627 - sev_classes_loss: 0.3530 - int_classes_loss: 0.2098 - sev_classes_accuracy: 0.9219 - int_classes_accuracy: 0.3906
Epoch 7/10
4/4 [==============================] - 1s 257ms/step - loss: 0.3066 - sev_classes_loss: 0.1193 - int_classes_loss: 0.1872 - sev_classes_accuracy: 0.9688 - int_classes_accuracy: 0.3906
Epoch 8/10
4/4 [==============================] - 1s 257ms/step - loss: 0.2522 - sev_classes_loss: 0.1043 - int_classes_loss: 0.1480 - sev_classes_accuracy: 0.9844 - int_classes_accuracy: 0.3906
Epoch 9/10
4/4 [==============================] - 1s 255ms/step - loss: 0.2455 - sev_classes_loss: 0.1221 - int_classes_loss: 0.1234 - sev_classes_accuracy: 0.9688 - int_classes_accuracy: 0.3906
Epoch 10/10
4/4 [==============================] - 1s 256ms/step - loss: 0.1855 - sev_classes_loss: 0.0848 - int_classes_loss: 0.1007 - sev_classes_accuracy: 0.9844 - int_classes_accuracy: 0.3906
```



Cost Dictionary and Functions

```
    exterior_location_class_map_cost = {
        "Front End (front bumper, hood, front grille, front fenders, front lights, front windshield)": 300,
        "Sides (doors, side mirrors, side fenders, windows, side skirts, pillars)": 200,
        "Rear end (rear bumper, trunk lid, tailgate, rear lights, rear windshield, rear spoiler)": 150,
        "Roof (roof panel, sunroof, roof rails)": 300,
        "Undercarriage (front and rear axles, exhaust system, chassis/frame)": 200,
        "Wheels and tires (alloy rims, wheel covers, tires)": 100,
        "Unknown": 0
    }

    exterior_damage_class_map_cost = {
        "Dent": 200,
        "Scratch/Scuff/Chip": 300,
        "Broken Glass": 100,
        "Headlight broken": 100,
        "Taillight broken": 100,
        "Smash": 300,
        "Rust": 300,
        "Puncture": 150,
```

```
        "Unknown": 0
    }

    exterior_severity_class_map_cost = {
        'No Damage': 0,
        'Minor Damage': 1,
        'Moderate Damage': 1.5,
        'Major Damage': 2,
        'Totaled': 3,
        "Unknown": 0
    }

    interior_severity_class_map_cost = {
        'No Damage': 0,
        'Minor Damage': 1,
        'Moderate Damage': 1.5,
        'Major Damage': 2,
        'Totaled': 3,
        "Unknown": 0
    }

    interior_location_class_map_cost = {
        'Dashboard Area (Instrument panel, center console, globe
compartment)': 1000,
        'Front Area (driver\'s seat, passenger seat, gear shift, pedals,
steering wheel and column, footwells)': 200,
        'Middle Area (rear seats, center armrest, floor console)': 125,
        'Storage and Utility (door pockets, cup holders, rear parcel
shelf, boot/trunk)': 200,
        'Roof Area (sun visors, rearview mirror, sunroof, roof lining,
grab handles)': 900,
        'Rear area (backseat pockets, rear air vents, rear center
console)': 550,
        'Miscellaneous (lighting, ports and outlets, controls)': 370,
        'Engine compartment (engine, radiator, battery, air filter, fuse
box)': 3100,
        'Transmission and Drivetrain (transmission, drive shaft,
differential, axles)': 1200,
        'Exhaust system (exhaust manifold, catalytic converter, muffler,
exhaust pipe)': 1150,
        'Fuel System (fuel tank, fuel pump, fuel injectors)': 950,
        'Brake system (brake pedal, master cylinder, brake pads, rotors,
brake calipers)': 900,
        'Suspension and steering (shock absorbers/struts, springs, control
arms, steering rack and pinion)': 2000,
```

```
        'electrical system (alternator, starter motor, wiring harness)':
1000,
        'climate system (air conditioning compressor, heater core, blower
motor)': 1300,
        'Body and frame (chasis, frame, body panels)': 600,
        "Unknown": 0
    }
```

These are arbitrary guesses for costs that are not based on any official source.

The formulas are:

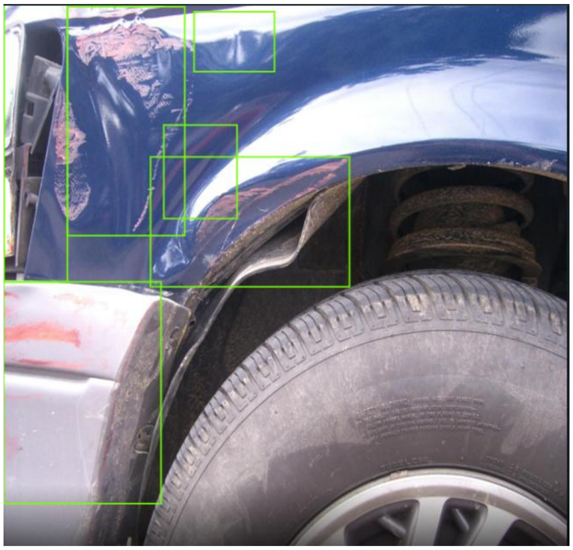
$$Exterior\ Cost = \big(sum(bounding\ box\ damage\ types) + sum(exterior\ locations)\big) * Exterior\ Severity$$

$$Interior\ Cost = Sum(interior\ locations) * Interior\ Severity$$

$$Total\ Cost = Exterior\ Cost + Interior\ Cost$$

Model Results vs Correct Answers

The pictures with red labels are the predictions from the model and the pictures with green boxes are manually marked.



Correct Box Labels: [Scratch/Scuff/Chip, Scratch/Scuff/Chip, Scratch/Scuff/Chip, Headlight Broken, Dent, Dent]

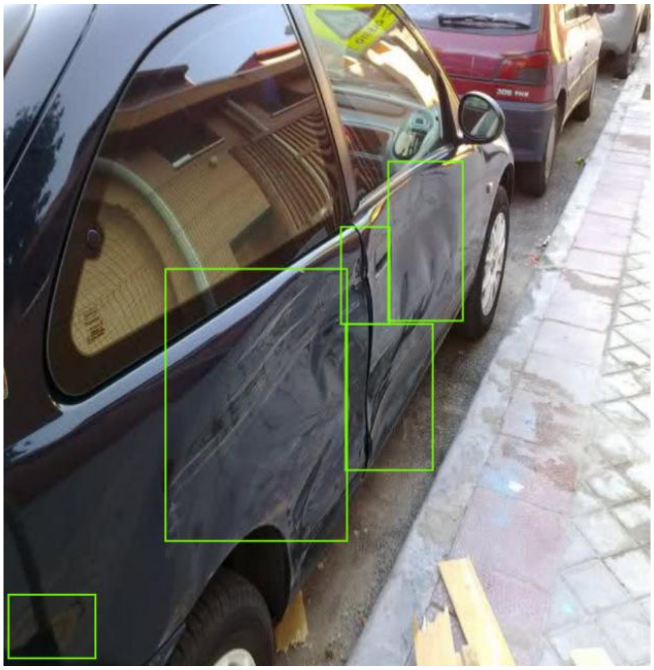
Correct Exterior Location: Front End (front bumper, hood, front grille, front fenders, front lights, front windshield)

Correct Exterior Damage Severity: Minor Damage

Correct Interior Location: Body and frame (chassis, frame, body panels)

Correct Interior Severity: No Damage

Correct Cost: Exterior (1700) + Interior (0) = Total Cost (1700)



Correct Box Labels: [Scratch/Scuff/Chip, Scratch/Scuff/Chip, Scratch/Scuff/Chip, Rust, Dent]

Correct Exterior Location: Sides (doors, side mirrors, side fenders, windows, side skirts, pillars)

Correct Exterior Damage Severity: Minor Damage

Correct Interior Location: [Storage and Utility (door pockets, cup holders, rear parcel shelf, boot/trunk), Body and frame (chassis, frame, body panels)]

Correct Interior Severity: Minor Damage

Correct Cost: Exterior (1600) + Interior (800) = Total Cost (2400)

Correct Box Labels: [Puncture, Puncture, Dent, Dent, Smash]

Correct Exterior Location: Sides (doors, side mirrors, side fenders, windows, side skirts, pillars)

Correct Exterior Damage Severity: Moderate Damage

Correct Interior Location: [Body and frame (chassis, frame, body panels), Suspension and steering (shock absorbers/struts, springs, control arms, steering rack, and pinion), Brake system (brake pedal, master cylinder, brake pads, rotors, brake calipers), Transmission and Drivetrain (transmission, drive shaft, differential, axles), Storage and Utility (door pockets, cup holders, rear parcel shelf, boot/trunk)]

Correct Interior Severity: Minor Damage

Correct Cost: Exterior (1800) + Interior (4900) = Total Cost (6700)

Correct Box Labels: [Scratch/Scuff/Chip, Scratch/Scuff/Chip, Dent, Dent]

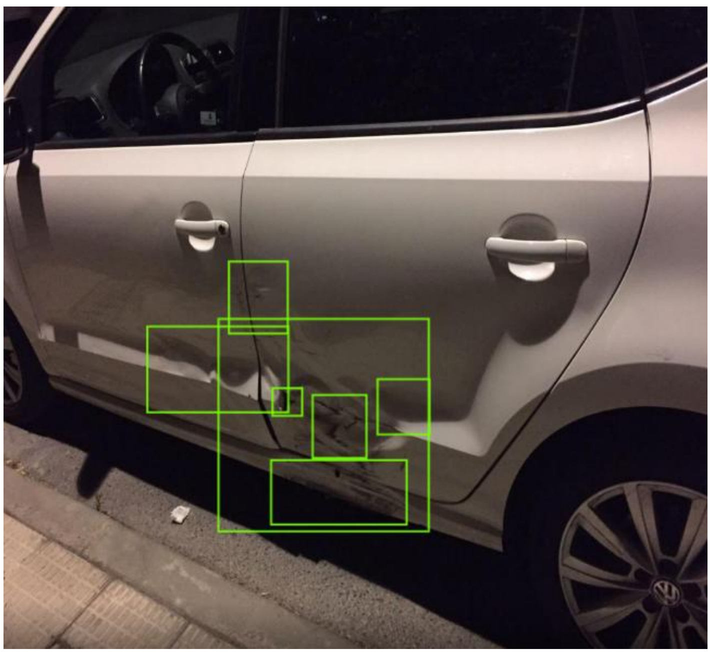
Correct Exterior Location: Sides (doors, side mirrors, side fenders, windows, side skirts, pillars)

Correct Exterior Damage Severity: Minor Damage

Correct Interior Location: Body and frame (chassis, frame, body panels)

Correct Interior Severity: No Damage

Correct Cost: Exterior (1200) + Interior (0) = Total Cost (1200)



Correct Box Labels: [Scratch/Scuff/Chip, Scratch/Scuff/Chip, Dent, Dent, Puncture, Puncture, Smash]

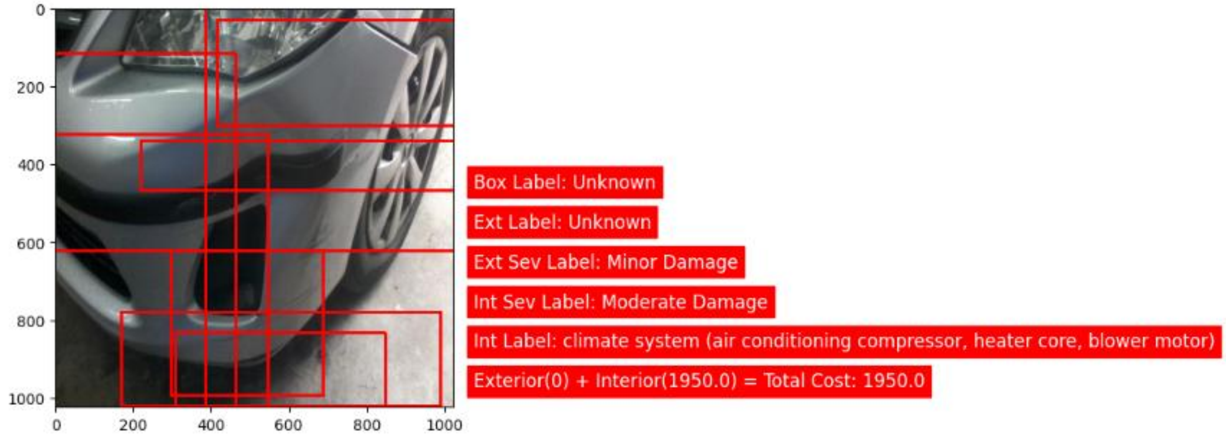
Correct Exterior Location: Sides (doors, side mirrors, side fenders, windows, side skirts, pillars)

Correct Exterior Damage Severity: Minor Damage

Correct Interior Location: Body and frame (chassis, frame, body panels)

Correct Interior Severity: No Damage

Correct Cost: Exterior (1800) + Interior (0) = Total Cost (1800)



Correct Box Labels: [Scratch/Scuff/Chip, Scratch/Scuff/Chip, Scratch/Scuff/Chip, Scratch/Scuff/Chip, Scratch/Scuff/Chip, Dent]

Correct Exterior Location: Front End (front bumper, hood, front grille, front fenders, front lights, front windshield)

Correct Exterior Damage Severity: Minor Damage

Correct Interior Location: Body and frame (chassis, frame, body panels)

Correct Interior Severity: No Damage

Correct Cost: Exterior (2000) + Interior (0) = Total Cost (2000)

Correct Box Labels: [Scratch/Scuff/Chip, Dent]

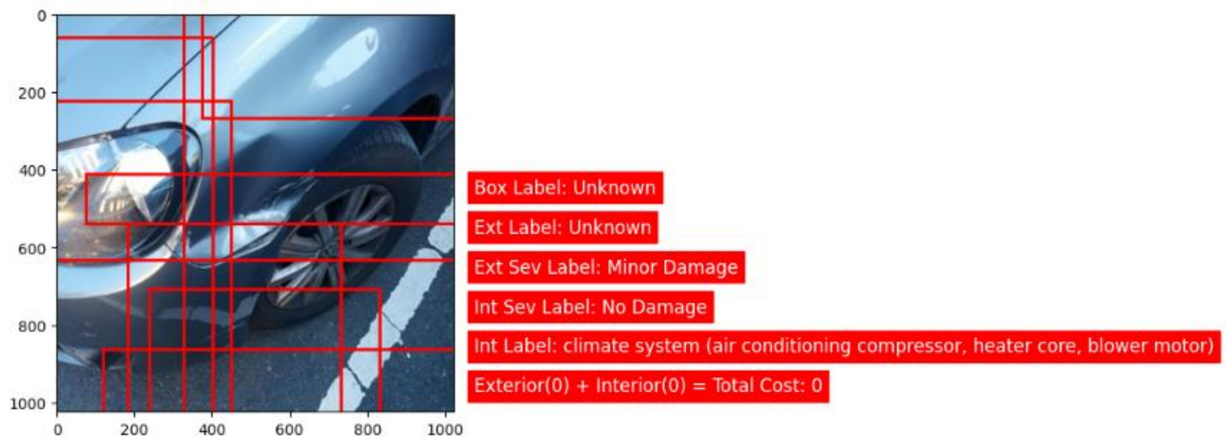
Correct Exterior Location: Sides (doors, side mirrors, side fenders, windows, side skirts, pillars)

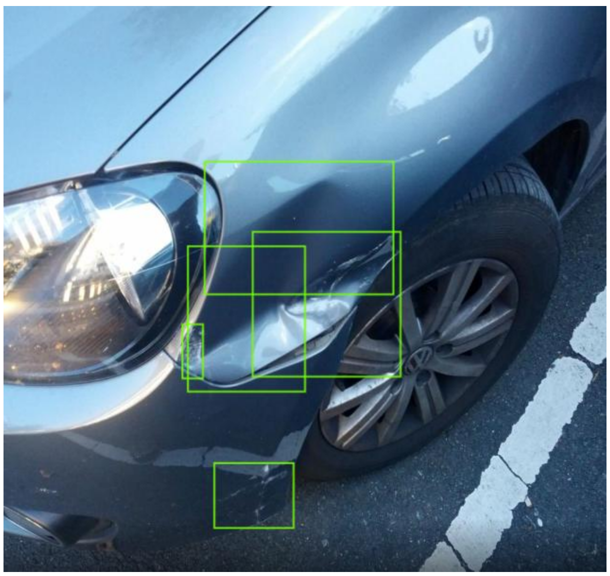
Correct Exterior Damage Severity: Minor Damage

Correct Interior Location: [Body and frame (chassis, frame, body panels), Storage and Utility (door pockets, cup holders, rear parcel shelf, boot/trunk)]

Correct Interior Severity: No Damage

Correct Cost: Exterior (700) + Interior (0) = Total Cost (700)





Correct Box Labels: [Dent, Dent, Scratch/Scuff/Chip, Scratch/Scuff/Chip, Scratch/Scuff/Chip]

Correct Exterior Location: Front End (front bumper, hood, front grille, front fenders, front lights, front windshield)

Correct Exterior Damage Severity: Minor Damage

Correct Interior Location: Body and frame (chassis, frame, body panels)

Correct Interior Severity: No Damage

Correct Cost: Exterior (1600) + Interior (0) = Total Cost (1600)