Master Thesis

Exploring Transfer Learning, Synthetic Data Augmentation, and Explainability Techniques for Improved Vehicle Damage Assessment

Louis Gauthy

Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science of Data Science for Decision Making
at the Department of Data Science and Knowledge Engineering
of the Maastricht University

Thesis Committee:

Dr. A. Briassouli (Internal Supervisor & Examinator)
Dr. M. Popa (Examinator)
C. Schleich (External Supervisor)

Maastricht University
Faculty of Science and Engineering
Department of Data Science and Knowledge Engineering

Jan 30, 2023

Acknowledgements

I would like to thank Dr. Alexia Briassouli and Ms. Corinne Schleich for their help and guidance throughout the research. They provided me with a valuable external view on my work. I would also like to thank NRB for hosting this thesis and providing me with valuable research equipment such as remote GPUs.

Abstract

This study aimed to improve the performance of a single-stage object detection model, Yolov5s, for vehicle damage detection. Various potential approaches, including transfer learning, image generation, and explainability analysis were explored to achieve this goal. The transfer learning experiments revealed that none of the models achieved comparable performance to the baseline model, which was trained solely on the Belgian company, Network Research Belgium (NRB) dataset and initialised with COCO pretrained weights. The use of synthetic image generation techniques as an innovative data augmentation approach was also explored, and the Stable Diffusion experiment results showed promising results. A used diffusion model was able to generate high-quality images of reconstructions of undamaged vehicles but also faced some limitations, which may be responsible for not improving the performances of the baseline model. Lastly, the explainability analysis provided further insights into the detection model limitations, which may be used for further performance improvement. Classifying damages was non-trivial due to the large inter-class similarity of damages, ambiguity in labelling, the large variety of damage shapes, and picture scenes. Overall, this study provides a valuable foundation for future research to further improve the model's performance in detecting damaged vehicles by, among others, addressing the limitations derived from the explainability study.

Contents

Introduction

Automatic vehicle damage assessment is an important topic for the vehicle insurance industry. Traditional insurance claim processes involve visual inspections of an insurance operator to evaluate the vehicle damages. This evaluation is subjective human interpretation and introduces a delay in the claim process. Patil, Kulkarni, Sriraman, et al. 2017 This delay is particularly problematic when insurance companies are overwhelmed with claims. Further, an accurate vehicle damage assessment model can reduce the claims' leakage of insurance companies, i.e. the gap between the optimal and actual settlement of a claim, thereby preventing the waste of financial resources of the insurance companiesReply 2013. This study is currently considering common discrete damage classes such as scratches and broken windows and it focuses on categorising and localising the damage. While the evaluation of damage costs remains a future challenge, the damage categorisation can already provide an indication of the potential costs. In more advanced settings, the model could be retrained to recognize more complex types of damage, and the tool could potentially be used by repair shops. The main goal of the study is the development of a framework towards a vehicle damage assessor by exploring the use of transfer learned damage detection models, the use of explainability techniques for those, and with the use of synthetically augmented datasets.

1.1 Context

The purpose of this study is to design and improve a framework that can automatically assess damage to a vehicle by analyzing images of the damages. This work is promoted by the Network Research Belgium (NRB) Belgian company and builds upon the previous research of J. de Bruyn Bruyn 2022. De Bruyn determined that Yolov5 is the most suitable model for the object detection task of detecting damages on vehicles. NRB has provided an internally constructed and labelled dataset of over four thousands of images 2.1 for model training use. However, NRB and Bruyn 2022 identified that the model is currently mainly limited by the quality of the training dataset. More specifically, there is a quality shortcoming in terms of image quantity, as four thousand images might sound little in comparison to the benchmark training datasets such as Zhuang, Lei, and Li 2020 20k images, Lin et al. 2014 328k or Everingham et al. 2010 19k. The other quality shortcoming comes from the fact that vehicle damage images are taken in an uncontrolled environment. This means that the images are taken by different people, with different cameras, in different lighting conditions, with different angles and different close-ups of the damages. Moreover, it has been observed that the damages are not always clearly visible since the internet-scraped pictures are often taken from a distance, in order to capture the whole accident scene. This makes the detection task more challenging since the model has to learn to detect damages in a variety of different conditions with a limited amount of training data.

1.2 Challenges in vehicle Damage Detection

The images provided by NRB have been scraped from the internet, resulting in a wide range of variations in the environment in which they are taken, see figure 2.1(b). In comparison, the Stanford Cars dataset offers a more consistent environment, see figure 2.2(b). It is more challenging for a model to detect an object from many different contexts and situations, as the model has not been exposed to sufficient examples of the object in those specific contexts during training. For example, a task in a well-defined context would be the detection of vehicle license plates on highway day camera images. Further, damage detection methods suffer from reflective metallic surfaces of vehicles that can be misclassified as damage, known as a false positive error Bagal n.d.

1.3 Problem Statement and research questions

1.3.1 Object Detection

The main goal of this study is to improve the performance of the detection model on damage detection. The previous research has already shown that the single-stage object detection model yolov5s is the most suitable for this task Bruyn 2022, in comparison to the Yolor and Detectron2 models. This study aims to explore ways of improving the performance of this model. One considered approach is to alter the model training process. In situations where the training data is limited, transfer learning, i.e. pretraining the model on a different dataset with a different task before training it on the main task, has been shown to be an effective method to consider. Three pretraining datasets are utilised, containing images of whole cars, and more images of damaged Cars. Based on this, the following first research question is formulated.

- *Does Transfer Learning help improving the model's performances?*

1.3.2 Image generation

One way to improve the performance of a machine learning model is to enhance the quality or quantity of the training data, rather than solely focusing on improving the model itself. This can be achieved through methods such as increasing the size of the dataset, selecting high-quality images, or selecting images that represent unique contexts. In this study, a more innovative approach is being explored, namely the use of synthetic image generation techniques to improve the data. Image generation technologies recently offered some impressive results. Generative Adversarial Networks (GANs) 2.4.1 and Stable Diffusion 2.4.2 have both demonstrated impressive results in this field and are being investigated in this study for their potential to generate new training images

It has previously been shown that estimating the undamaged state of a vehicle using laser scans can improve the performance of models classifying 3D laser scans of damaged vehicles. Inspired by this finding, this study aims to examine whether including estimates of undamaged vehicles can improve the performance of a model for detecting vehicle damage in images, instead of laser scans. The following two research questions emerge from this investigation The two following research questions emerge from this.

- *How to generate an undamaged version of a damaged vehicle image?*
- *Which metrics, best evaluates and separates an undamaged vehicle from a damaged vehicle?*

1.3.3 Explainability

Investigating why and how a machine learning model made a decision is as important as having a model making accurate decisions. This type of explainability study increases trust towards the model's decisions and transparency. It can also help to diagnose wrong decisions, and this diagnosis is valuable feedback to further improve the model. The final section of this study is dedicated to explainability analysis, and includes the following last research question.

- *To what extent can we explain the decisions of a damaged vehicle images detector?*

1.4 Related work

1.4.1 Car Damage Assessors

Except for Jayawardena 2013 who proposed to detect vehicle scratch damage by registering a 3D CAD model of undamaged vehicles (ground truth) on the image of the damaged vehicle, the latest research on vehicle damage assessment are mainly using a transfer learning approach with a convolution neural network (CNN) based model for 2D images of damaged vehicles Patil, Kulkarni, Sriraman, et al. 2017, Patil, Kulkarni, and Karande 2018, Fouad et al. 2022. For instance, Fouad et al. 2022 utilised the DenseNet169 and VGG19 pretrained networks to classify damages by severity and location.Patil, Kulkarni, Sriraman, et al. 2017 also explored the approach with 6 different transferred models pretrained on the ImageNet dataset Deng et al. 2009. Both Fouad et al. 2022 and Patil, Kulkarni, Sriraman, et al. 2017 also experimented with an ensemble learning approach, combining the predictions of the different individual models. However, the obtained results might not be significant enough to show the superiority of the approach, with an increase of 2% in F1 score in Fouad et al. 2022 and an increase of 1.29% from the most performant model in Patil, Kulkarni, Sriraman, et al. 2017. The damage classes used in these studies varied but related to the location, severity, or generic type of damage. For example, Patil, Kulkarni, Sriraman, et al. 2017 considered bumper dent, scratch, door dent, glass shatter, broken headlights, broken tail lights, and smashed, while Fouad et al. 2022 focused on broken glass, broken headlights, damaged tail lights, and dents classes. Ruitenbeek and Bhulai 2022 considered bend, bump, cover damage, crack, dent, glass shatter, hail, light broken, rust, scratch, and tire crack, and Dwivedi et al. 2019 considered bumper dent, scratch, door dent, glass shatter, head-lamp broken, tail-lamp broken, smashed. Further, Patil, Kulkarni, Sriraman, et al. 2017 and Dwivedi et al. 2019 used an 'undamaged' class. Even though those class sets are similar, it must be noted that the classification task is influenced by intra-class interference. That choice of classes may influence the difficulty of the classification and thus the performances of the models. Ruitenbeek and Bhulai 2022 showed that more detailed object categories result in less inter-class interference for damage detection.

The best-reported accuracies in these studies were 89% for Patil, Kulkarni, Sriraman, et al. 2017, using about 1250 training images, and Dwivedi et al. 2019 which reported an accuracy of 96%, using about 1000 training images. All studies employed some data augmentation techniques.

Despite their promising results, Dwivedi et al. 2019 pointed out that a larger dataset could improve the results and make the system more robust for real-life scenarios. Additionally, Ruitenbeek and Bhulai 2022 found that external data extracted from Google image search for training enhances the detection performance on the internal dataset. The field of study is generally impacted by the lack of benchmark data. All considered studies went through the data collection and annotation processes. Fouad et al. 2022 data was scraped from the Google, Bing, and DuckDuckGo search engines using a web scraper and Patil, Kulkarni, Sriraman, et al. 2017, Dwivedi et al. 2019 collected the data through the web crawler.

In addition to classification, some studies have also explored the problem of damage localisation, or identifying the specific damage location area on the vehicle. Yolo and its variants have been widely used for the task of object detection, which involves object localisation, and have achieved promising results. In the damage detection field, Ruitenbeek and Bhulai 2022 obtained a mean average precision (mAP) of 0.33, and Dwivedi et al. 2019 reported an mAP of 0.7423 using a (416x416) resolution. Ruitenbeek and Bhulai 2022 also explained the challenges of damage localisation by the fact that it is a non-trivial task due to the different shapes of the damages.

1.4.2 Image to image translation

Image-to-image translation (IIT) refers to the process of generating a new image that is a modified version of a given input image. A generative model learns the mapping between a category X and Y of images. The wide range of applications of IIT includes object synthesis, style transfer, and data augmentation. This study will use the IIT techniques as data augmentation techniques.

Deep learning experienced a major breakthrough when Ian J. Goodfellow introduced GANs in 2014 Goodfellow et al. 2014a, which revolutionised the image generation field. GANs have been shown to be promising in different fields, including image generation, and IIT. They were shown to be effective at generating high-quality synthetic data. GANs are known to be difficult to train. It is important to carefully design and tune the generator and discriminator networks and the training process to achieve good results. In the IIT application, J.-Y. Zhu et al. 2017 introduced a variant of GANs called Cycle-Consistent GAN or CycleGAN, which uses a cycle consistency loss to enforce that $F(G(x)) \approx x$ and $G(Y(y)) \approx y$ where G and F are the domain mappings between images categories X and Y. J.-Y. Zhu et al. 2017 illustrated the CycleGAN capabilities by converting horses to zebras, and summer to winter pictures. Then, Isola et al. 2016 introduced Pix2Pix GANs. This variant of GANs is noteworthy for its use of a PatchGAN discriminator which is designed to evaluate the realness of small patches of an image rather than the entire image, combined with the use of a U-Net generator architecture. Pix2pix was illustrated by an application of Satellite pictures conversion to Google maps images.

Another approach to image-to-image translation is the use of autoencoders Yoo, Eom, and Choi 2019. Autoencoders are neural networks trained to reconstruct their inputs, and can be used for IIT by training them on pairs of input and output images. However, datasets containing those image pairs are a rare commodity J.-Y. Zhu et al. 2017.

Lastly, Diffusion models have gained significant popularity in recent years due to their ability to generate high-quality images Saharia, Ho, et al. 2021, and their ability to overcome state-of-the-art GANs Dhariwal and Nichol 2021, as well as their benefits of not requiring adversarial training and being scalable. Saharia, Chan, Chang, et al. 2021 put emphasis on the fact that no task-specific architecture customization, nor changes to hyper-parameters or the loss were needed to tune the model on different tasks such as inpainting and colourisation. These models are based on the idea of iteratively updating the pixels of the output image through a series of diffusion steps, guided by the input image and a set of constraints. More specifically, the input images are diffused toward

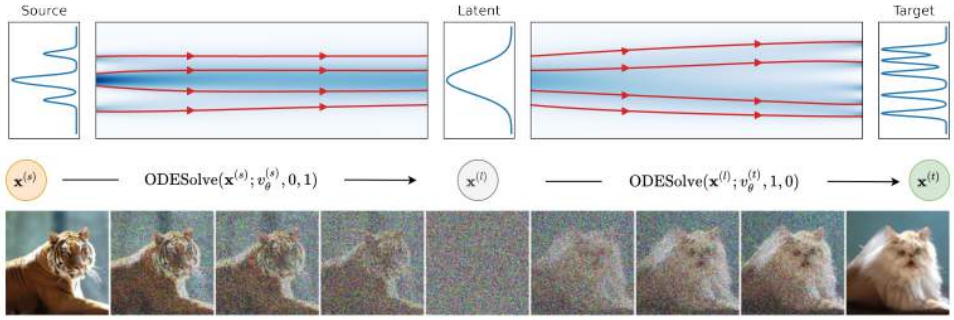

Source
Latent
Target
$\mathbf{x}^{(s)}$
$\mathrm{ODESolve}(\mathbf{x}^{(s)}; v_\theta^{(s)}, 0, 1)$
$\mathbf{x}^{(l)}$
$\mathrm{ODESolve}(\mathbf{x}^{(l)}; v_\theta^{(t)}, 1, 0)$
$\mathbf{x}^{(t)}$

Figure 1.1: Illustration of an image-to-image translation using diffusion models Su et al. 2022, showing the gradual diffusion process (image noising) and the gradual image generation process (image denoising)

a standard Gaussian distribution and the Diffusion model then converts them back into samples from an empirical data distribution through an iterative denoising process.

1.4.3 Explainability for Damage Detection

In the paper cited here above Patil, Kulkarni, Sriraman, et al. 2017, Patil, Kulkarni, and Karande 2018, Ruitenbeek and Bhulai 2022, Fouad et al. 2022, only Dwivedi et al. 2019 included a section dedicated to model explainability. They used class activation mappings (CAM) to visualise which pixels were responsible for predicting a specific class. Their results allowed us to visualise that their model was generalising well by observing that the damaged regions of the images are often highlighted. With the recent blossoming of deep learning methods that are gradually included in real-life applications, raised from a need to explain the models' decisions in a human-comprehensible way in order to gain trust in these deeply complex algorithms. That boosted the explainability field of research and gave rise to a large variety of explainability technologies Ras et al. 2020. This study focuses on the three following ones: SHAP, LRP, and Grad-CAM.

1.5 Study Plan

In this study, object detection tasks are being carried out using the Yolov5s model, as the previous research of de Bruyn Bruyn 2022 on the topic has confirmed that yolov5 outperforms the other models such as FRCNN and YoloR in the vehicle damage detection task. Yolov5 was at that time the state-of-art and new updates of the model such as Yolov7 are now available. However, in this study, the Yolov5s model is kept as the unique considered model for the object detection task. This allows for a fair comparison with the previous research of de Bruyn Bruyn 2022 on the topic, and limits the scope of the study to the exploration of different para-model improvement methods, such as data or model training improvement, rather than exploring improved versions of the model which are theoretically expected to perform better, and which could be explored in a future study. To improve the model training, transfer learning followed by fine-tuning is performed on the combinations of four different datasets having their own specific different detection task. To improve the data, synthetic image generation is tested to expand the dataset, namely GANs and

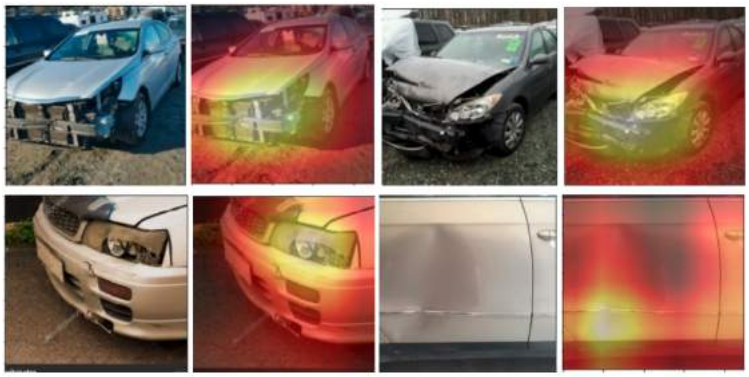


Figure 1.2: Examples of CAMs, obtained in Dwivedi et al. 2019, showing that the damaged regions of the images are often highlighted as the discriminative image regions used by the model to identify that category

Stable Diffusion. Lastly, as an explainability analysis, the study applies three different techniques for computing a heatmap visualisation of gradients or the model attention.

Methods

2.1 Datasets

In this section are introduced the four datasets used in this study. A summary of their characteristics is provided in 2.1

Original NRB dataset, NRB This is the baseline dataset of 4373 images of damaged cars, provided by the Belgian company NRB. The images were scraped on the internet and the classes and class location labels were annotated by the company. It is a multiclass dataset and classes inform on the location and the severity of the damage. The classes are the following: Broken glass, Major front damage, Major rear damage, Major side damage, Minor front damage, Minor rear damage, Minor side damage, Scratches. The data is somehow balanced. It can be observed in 2.1(a) that the class 'Major front damage' is overrepresented with about 800 samples while all the other classes are represented by 500 to 300 samples.

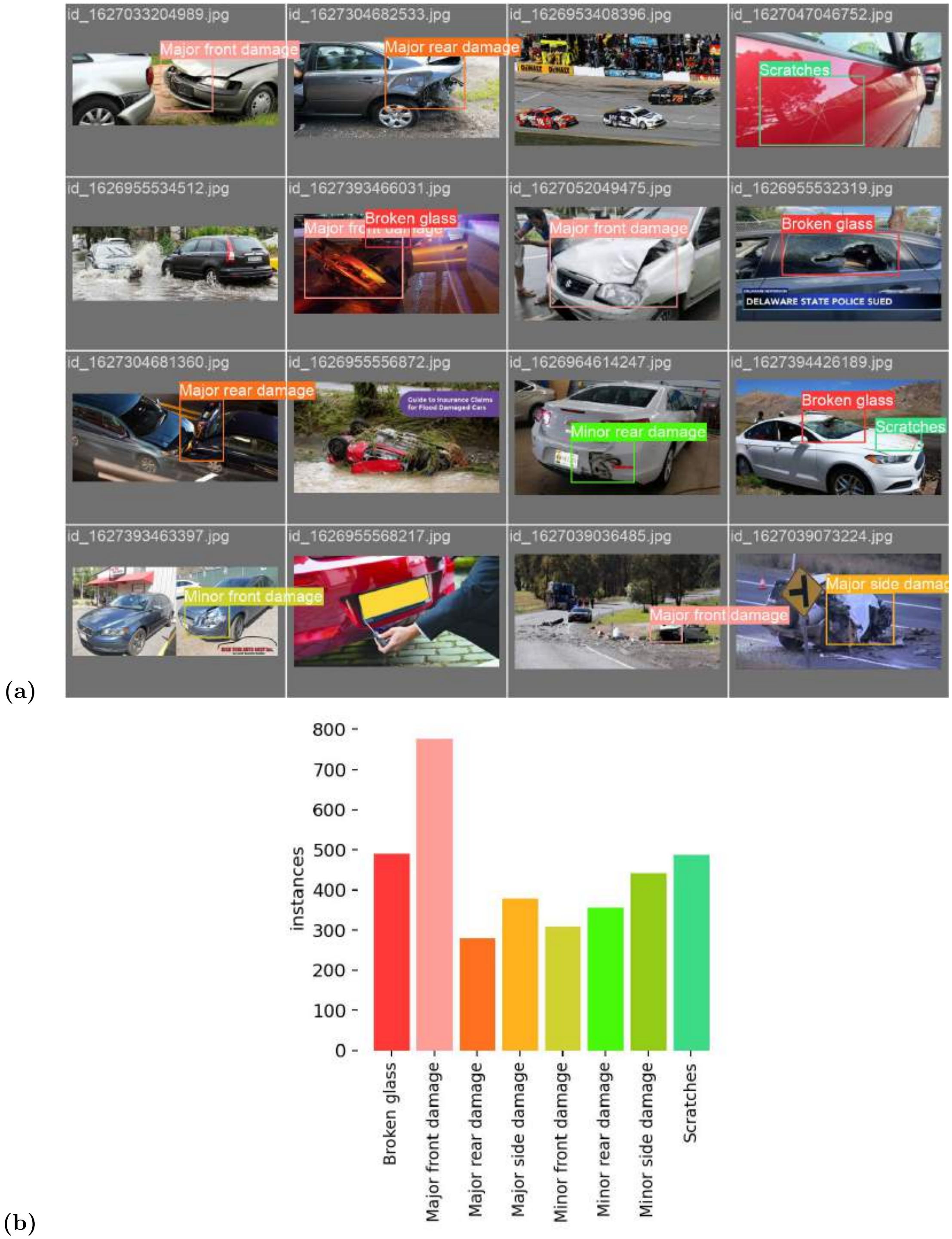


Figure 2.1: (a) count plot of the classes of the NRB dataset, (b) a batch sample of the NRB dataset

Stanford Cars This large-scale dataset contains 16 185 whole cars and 196 classes which are cars model and year Krause et al. 2013. The images in the dataset are high-resolution and are annotated with the location of the car in the image. The data is balanced and approximately 40 instances of each class are contained. The Stanford Cars dataset has been widely used in research on deep learning and has been used to train and evaluate numerous models Cubuk et al. 2019 Touvron, Cord, Sablayrolles, et al. 2021 Graham et al. 2021 Tatsunami and Taki 2022 Touvron, Cord, Douze, et al. 2020. It has also been used as a benchmark for comparing the performance of different models on the task of car detection and classification, it is thus beneficial to include it in this study.

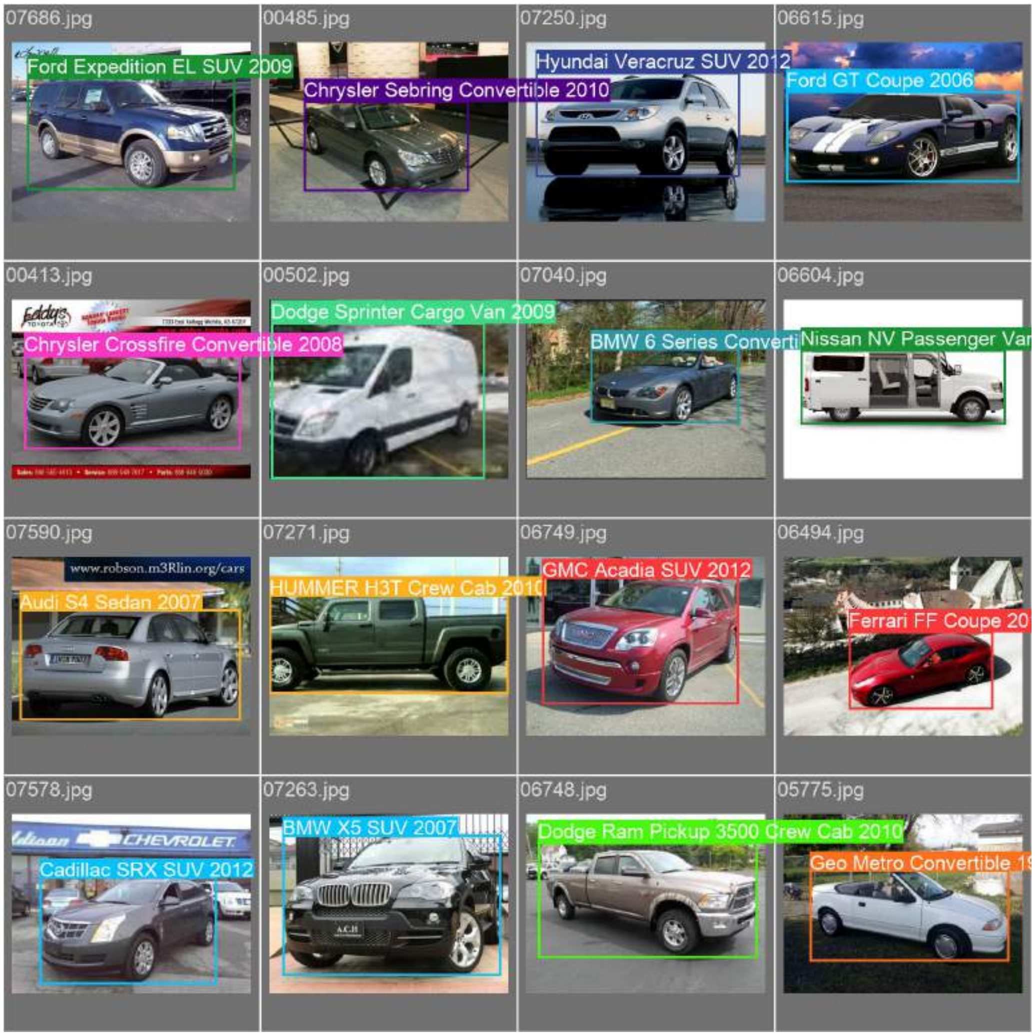


Figure 2.2: a batch sample of the Stanford Cars dataset

Damaged Vehicle Images Dataset, Roboflow This damage detection dataset containing 5072 augmented images from 1001 original images is hosted in Roboflow Sammy 2022. It contains similar classes as the baseline nrb dataset, for example 'Scratches' and 'slight scratch' or 'severe scratch'. The classes included in this dataset are: crack and hole, medium deformation, severe deformation, severe scratch, slight deformation, slight scratch, and windshield damage. It be observed in 2.3(a) that the classes are almost balance, with about 800 instances per class. The class 'slight scratch' is overrepresented with about 1750.

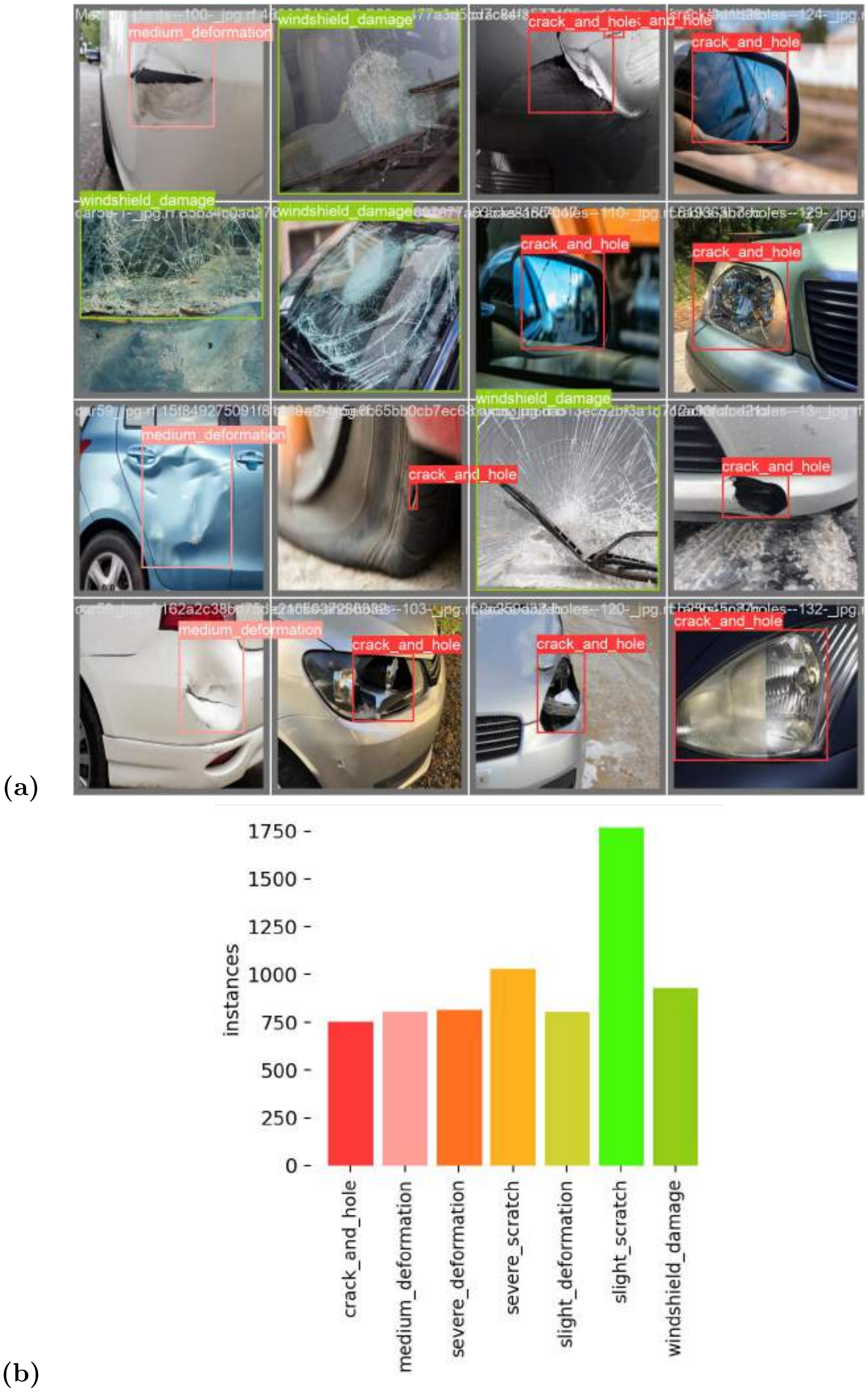


Figure 2.3: (a) count plot of the classes of the Roboflow dataset, (b) a batch sample of the Roboflow dataset

Damage Image Dataset, NEOKT This last dataset includes data scraped from Google Images using Selenium, hand-labelled for classification Neokt 2017. It contains 1150 good-quality images of damaged vehicles, with types of labelled classes (severity and location). It is also supplemented with the same amount of whole images coming from the Stanford Car Image Dataset. Note that this data is a classification dataset, no bounding box labels are included. The dataset is then converted into an object detection dataset, see 3.1 for the conversion process details. The different classes included in this dataset are: "damaged", and "whole". The classes are balanced.

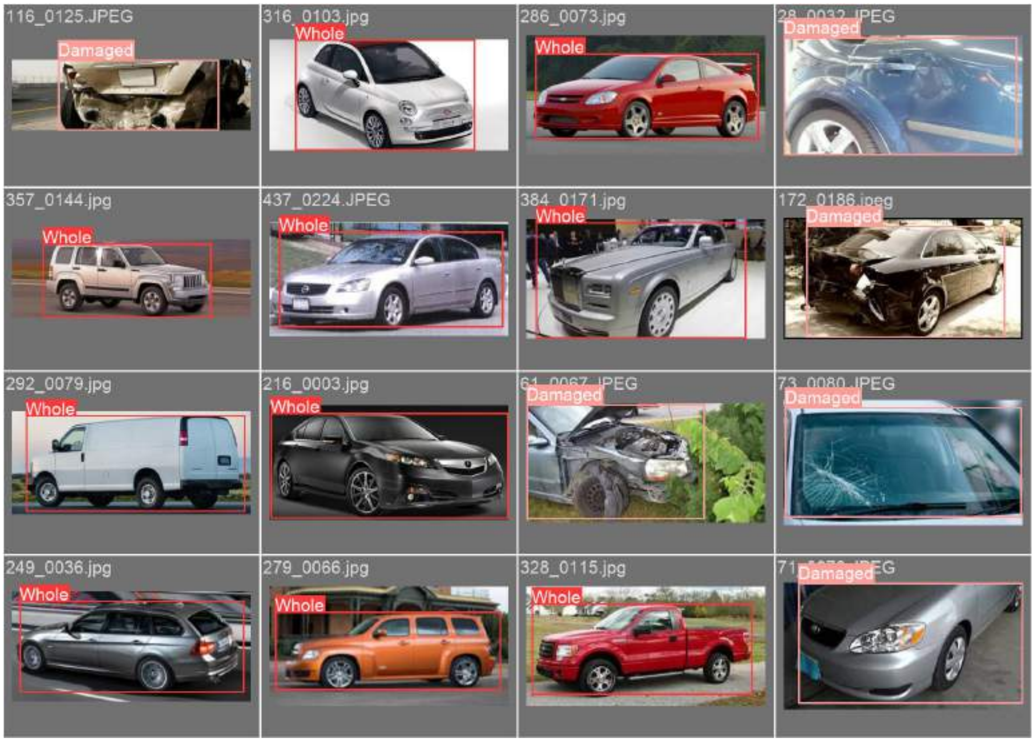


Figure 2.4: a batch sample of the NEOKT dataset

2.2 Single-stage object detection

In contrast to two-stage object detection, single-stage deep learning models like YOLO or You Only Look once Jocher et al. 2022 aim to incorporate the object localisation and detection tasks with one single model. This strategy is known to be effective for online detection tasks. Yolov5 is the current state-of-the-art in the object detection field, due to its high speed and accuracy.

Yolov5's architecture is composed of three main building blocks. The Backbone is responsible for extracting meaningful features from an input image. The Neck processes the extracted features to create a feature pyramid. For Yolov5, the Backbone is mainly composed of the CSPDarknet53 model. Those pyramids make the model more robust to changes in the scale and shape of objects. The Head performs the final detection step by applying box anchors on the features pyramids and

Name	**Size**	**Task**	**Source**
NRB	4373 images	Damage detection	NRB (scraped from internet)
Stanford Cars	16185 images	Car detection and classification	Krause et al. 2013
Roboflow	5072 augmented images from 1001 original images	Damage detection	Sammy, Roboflow
NEOKT	1150 images	Classification	Neokt 2017 (scraped from Google Images)

Table 2.1: Summary of datasets

returns the class probabilities, objectness scores, and bounding boxes. Bounding boxes and anchor boxes are both rectangle shapes are used to localize objects, but anchor boxes have predefined sizes and aspect ratios.

2.2.1 Metrics

Four metrics are used to evaluate the performance of the detection models. Firstly, the Recall is given by the following formula: $Recall = TP/(TP + FN)$. In plain English, the recall is the total number of true positives detected by a model, in comparison to the total number of true positives that should have been detected by the model. Secondly, the Precision is given by the following formula: $Precision = TP/(TP + FP)$. In plain English, the precision is the total number of true positives detected by a model, in comparison to the total number of true positives that the model has detected. In other words, the precision gives information on the confidence of the model in its predictions. Thirdly, the F1 score is calculated by the formula: $F1 = 2 * \frac{precision*recall}{precision+recall}$. In other words, the F1 score is a metric summarising the precision and the recall. F1 ranges from 0 to 1, 0 being the worst and 1 the best score. Finally, the mean Average Precision (mAP) metric is widely used for evaluation of object detection algorithms. It is calculated as the average between the Average Precisions of the different object classes at a minimum Intersection over union (IoU). Where IoU is the ratio between the area of overlap and the area of union between the target a predicted area. For this study, the IoU threshold is set to 0.5, the official threshold of VOC Everingham et al. 2010. Further, the Average Precision is defined as the area under the Precision/Recall curve H. Zhu et al. 2020. Thus, AP is the average of the precision values at a set of various recall values.

2.3 Transfer Learning

The main idea behind the transfer learning training strategy is to use a pretrained model, i.e. a model that has been trained for some task, and to continue training it on a different task by allowing uniquely the topmost layers to learn. The bottom layers are responsible for the feature extraction, whereas the top layers are responsible for making the classification and detection of the extracted feature. The method thus involves retraining a classifier/detector of a model while reusing its feature extractor. Transfer Learning is useful in situations where there is limited data available for the new task, as it typically requires fewer training images and fewer training epochs. By using a pre-trained model, some knowledge of the original task is transferred to the new model, which can improve its performance on the new task. In general, any pre-trained network can be used for transfer learning, but the more similar the two tasks are, the more effective the transfer learning will be.

Fine-Tuning is a training technique similar to Transfer Learning that can be applied after or instead of it. In comparison to Transfer Learning, the process involves unfreezing some of the topmost feature extraction layers and typically reducing the learning rate. By allowing these layers to continue learning and adjusting their weights, the model can improve its feature representations towards the new task and achieve higher model performance Patil, Kulkarni, and Karande 2018. The method is to be used cautionary since it may cause model overfitting, especially when the transferred dataset is small and is similar to the original.

2.4 Synthetic image generation

One challenge addressed in this study is to expand the current training dataset using synthetically generated images. To do this, the study proposes to compute an estimate of the undamaged version of each damaged vehicle image in the training dataset. By pairing each damaged image with an estimate of its undamaged counterpart, it is hoped that the model can better understand the features of damaged vehicles and improve its ability to detect damage. The approach could be seen as an advanced data augmentation method as it creates new training data by transforming existing images

2.4.1 GANs

Generative Adversarial Networks (GANs) are generative models composed of two different neural networks, a convolutional neural network, called a generator and a transposed convolutional neural network, called a discriminator Goodfellow et al. 2014b. Those two networks are trained together to generate new synthetic data that is similar to a training dataset. Formally, GAN is defined as follows: the generator G is a neural network that takes as input a noise vector z and generates a synthetic sample x = G(z). The discriminator D is a neural network that takes an image sample x as input and outputs a probability p = D(x) indicating the likelihood that x is a real sample from the training dataset rather than a synthetic sample generated by the generator. The generator and discriminator are trained to compete against one another: the generator aims to produce synthetic samples indistinguishable from real samples, while the discriminator aims to correctly classify real samples from synthetic samples. In other words, the generator is then trained to mislead the discriminator, while the discriminator is trained to unmask the generator's creations. This training process has been shown to be effective in generating synthetic samples that are similar to a training dataset, but it can also be unstable. The art of GAN training is to be able to find a balance between the training of both neural networks. If the discriminator learns too quickly, every generator attempt to fool the discriminator will be fruitless and the generator may be unable to improve its synthetic samples. Conversely, if the generator learns too quickly and fools immediately its competitor, the discriminator may not be able to provide feedback to the generator to improve its creations.

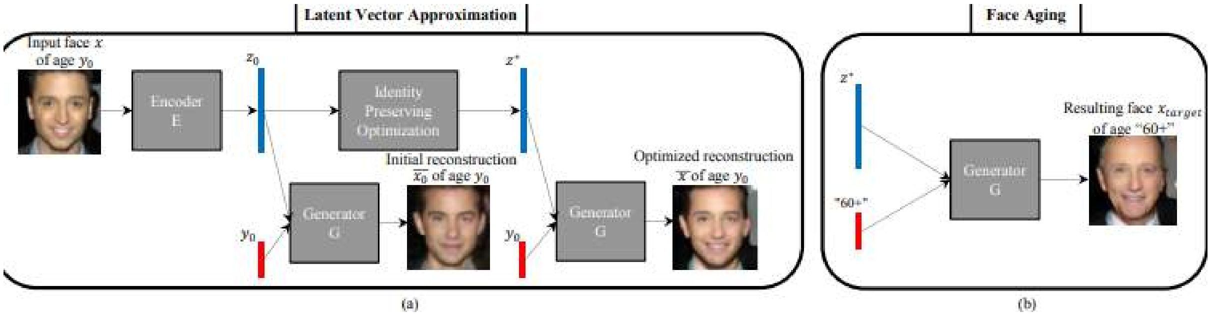

Latent Vector Approximation
Input face x of age y_0
Encoder E
z_0
Identity Preserving Optimization
z^*
Initial reconstruction $\overline{x_0}$ of age y_0
Generator G
y_0
Optimized reconstruction $\overline{x}$ of age y_0
Generator G
y_0
(a)
Face Aging
z^*
"60+"
Generator G
Resulting face x_{target} of age "60+"
(b)

Figure 2.5: Illustration of the face-ageing GAN pipeline, developed in Antipov, Baccouche, and Dugelay 2017, where (a) represents the latent vector optimisation of the pipeline, where the Encoder E is optimised to map the Image space with the vector space, and E is further optimised with the Generator G on identity preservation constraints. (b) represents the preliminary pipeline step on training a Generator on a face-ageing task with an age category condition

The present study investigates the use of an age-conditional GAN (acGAN) architecture, as proposed in Antipov, Baccouche, and Dugelay 2017, for the task of damage detection on vehicle images. This architecture, illustrated in Figure 2.5, was originally designed for the task of face ageing transformation, and this study thus investigates the transfer of a face ageing task to the vehicle "undamaging" task. The approach described in Antipov, Baccouche, and Dugelay 2017 involves two steps: first, finding an optimal latent vector that allows for the reconstruction of an input face (or vehicle) image as closely as possible while preserving the original person's or (vehicle's) identity 2.5-(a), and second, using the latent vector to generate a resulting face (vehicle) image at a target age (target damage severity) by switching the age (damage) condition at the input of the generator 2.5-(b). In addition to the acGAN generator and discriminator models, the methods involve using and training of an encoder network 2.5-(a) used for approximatively reconstructing an input face image while preserving the original person's identity. The encoder maps an input image to a latent vector in the noise space, which is then fed into the generator of the acGAN model along with the target age condition. The generator is trained to produce a reconstructed face image similar to the input image, and the encoder to minimize the difference between the input image and the reconstructed face image.

2.4.2 Stable diffusion

The second image generation technique explored in the study is diffusion-based text-to-image generation. More specifically, a latent diffusion model that uses a fixed, pretrained text encoder is used, suggest from the Imagem paper Saharia, Chan, Saxena, et al. 2022 and implemented in Rombach, Blattmann, Lorenz, Esser, and Björn Ommer 2022. Latent diffusion models which are variations of diffusion models were introduced in Rombach, Blattmann, Lorenz, Esser, and Bjorn Ommer 2022. Diffusion models are described in Rombach, Blattmann, Lorenz, Esser, and Bjorn Ommer 2022 as "probabilistic models designed to learn a data distribution p(x) by gradually denoising a normally distributed variable, which corresponds to learning the reverse process of a fixed Markov Chain of length T". In other words, the key principle behind diffusion models is to "decompose an image generation process into a sequential application of denoising autoencoders" Rombach, Blattmann, Lorenz, Esser, and Bjorn Ommer 2022. The denoising autoencoder architecture is traditionally the U-Net convolutional network architecture. In comparison to GANs, there is now only one type of model to be trained, the U-Nets, and diffusion models are then exempted from the tedious adversarial training. Further, the task of gradually restoring small amounts of Gaussian noise to be achieved by the U-Nets is a lighter and more practical task in comparison to Generators which are asked to map the input noise space to the output image space in a single attempt. In comparison to diffusion models, latent diffusion models are performing noise reduction in a compressed latent space of lower dimensionality. This dimensionality reduction is performed using autoencoding models that "learn a space that is equivalent to the image space but offers reduced computational complexity" Rombach, Blattmann, Lorenz, Esser, and Bjorn Ommer 2022. The diffusion technology is constrained by its training computational high costs, hundreds of GPU days for the best models. The costs derive from the denoising processes that are performed in the pixel space. Latent diffusion models address this shortcoming and require cheaper and faster training. Overall, the use of latent diffusion models represents a promising and efficient alternative to GANs, with the advantageous functionality of allowing to incorporate textual information into the synthesis process.

2.5 Synthetic image evaluation

Assessing the quality of synthetic images is a complex task, as it can be difficult to establish clear criteria for determining the realism or high quality of an image, which is often left to subjective human judgment Borji 2018. To date, no consensus has been made on a benchmark evaluation metric to use. The challenges behind the evaluation task derive from multiple factors: the evaluation of the diversity of the images, the lack of image datasets humanly labelled with scores assessing the image quality, and the fact that the definition of image "quality" is left to human subjectivity, thus it is unclear how one should judge and quantify image quality.Borji 2018. Nevertheless, it is always helpful to involve human intervention in the image quality assessment process.

In order to guide and train the image generation models introduced in 2.4, the following metrics are utilised.

2.5.1 Frehet inception distance

Frechet Inception Distance (FID) is a measure of the distance between two distributions of images. It was introduced by Heusel et al in Heusel et al. 2017 and is now with the Inception Score among the most popular metrics for image quality evaluation. Formally, having two datasets, A of real images and B of generated images, the activations of a pre-trained Inception network from these datasets are extracted, $A = a_1, a_2, ..., a_m$ and $B = b_1, b_2, ..., b_n$. These activations can be seen of as a compact and meaningful representation of the images. The Inception network is a convolutional neural network trained on the ImageNet dataset. FID is defined as follows:

$$FID(A, B) = ||\mu_A - \mu_B||^2 + Tr(\sigma_A + \sigma_B - 2(\sigma_A \sigma_B)^{\frac{1}{2}}$$

where $Tr(X)$ is the trace of the matrix X, μ_A, σ_A and μ_B, σ_B, the mean and covariance matrices of the normal distribution of the images sets A and B, respectively Detlefsen et al. 2022.

A lower FID score indicates a proximity in the distributions of A and B in terms of Inception high-level features, suggesting that the generated images are similar to the real images. In this study, FID was selected over the Inception Score, another popular image quality evaluation metric, due to its demonstrated alignment with human judgments, greater robustness to noise compared to Inception Score Borji 2018 (e.g., a negative correlation between FID and the visual quality of generated samples), and due to its ability to detect intra-class mode collapse (e.g., models generating too similar images may score highly on IS but poorly on FID), and due to its greater sensitivity to various types of artefacts in images Borji 2018.

2.5.2 Vehicle Resemblance Score

In the study, a new metric called Car Resemblance Score (VRS) is introduced. This metric is designed as an image quality metric specialised in damaged/whole vehicle image evaluation. VRS is computed using a ResNet50 He et al. 2015 convolutional neural network pretrained on Imagenet dataset Deng et al. 2009 and fine-tuned on both Stanford Cars Krause et al. 2013 and NRB dataset 2.1. The classification task of the model is set to classify images of damaged vehicles, provided in the NRB dataset, and images of undamaged vehicles, provided by the Stanford Cars dataset. The target classes are "damaged" and "whole". The model is trained for 15 epochs with a learning rate of $1e^{-3}$. The VRS scores are determined by the output activation of the class "whole".

2.5.3 Structural Similarity Index Metric

The structural Similarity Index Metric (SSIM) was introduced in Wang et al. 2004. It is a metric used to measure the similarity between two images. It is used in image processing and computer vision applications, and it has been shown to be effective at quantifying the similarity between images. SSIM is defined as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where x and y are the two images being compared, $\mu_x\sigma_x, \mu_y\sigma_y$ are the means and standard deviations of the pixel intensities in x and y, respectively. $\sigma_x y$ is the covariance of the pixel intensities in x and y, and c_1 and c_2 are constants are used to stabilize the division.

In other words, two images are similar if they have similar means, variances, and pixel intensities covariances. SSIM is designed to be a more robust measure of image similarity than simple pixel-wise comparison methods such as mean squared error (MSE) Monsters 2020W and it ranges from -1 to 1, with higher values indicating greater structural similarity between the two images. Lastly, it is worth noting that the SSIM index is only one statistical measure of image similarity, and it may not always accurately reflect the perceived similarity of the images by humans Monsters 2020W.

2.5.4 Vehicle Similarity Score

Evaluating the similarity between two images is a challenging task, as it is often difficult to define a clear criterion for what makes an image "similar" or "dissimilar". To address this challenge, in addition to SSIM, a new metric called vehicle similarity score (VSS) is introduced. This metric is designed to measure the similarity between images of damaged and undamaged vehicles, especially. Inspired by FID, VSS also takes advantage of convolution neural networks' high-level features. It is computed using a Vision Image Transformer (ViT) Wu et al. 2020 model pretrained on Imagenet dataset Deng et al. 2009 used and fine-tuned on the Stanford Cars dataset Krause et al. 2013 by S. Sridhar Sridhar78 2021. The fine-tuning task is a vehicle brand classification task and the model is fine-tuning for 4 epochs with a learning rate of $2e^{-5}$. The feature representations of the first to last layer are used. They produces two sets of activations: $A = a_1, a_2, ..., a_m$, for the real damaged vehicle image, and $B = b_1, b_2, ..., b_n$, for the generated undamaged vehicle. VSS is calculated as follows:

$$VSS(A, B) = -\frac{\sum |A - B|}{max(\sum |A|, \sum |B|)}$$

where $|x|$ is the absolute value of x. This equation calculates the sum of absolute gradient differences, normalised by the total gradient absolute weight.

2.6 Explainability for deep learning

In order to provide insights into how the models are making decisions or predictions, three different visualization techniques are utilised.They will help understand how the model is processing the input data and making predictions.

2.6.1 SHAP Values

SHAP (SHapley Additive exPlanation) values analysis is a make model-agnostic explainability technique, which makes it suitable to explain deep learning models that are complex. It is a statistical method that investigates the influence of input feature variations on the model's outcomes by average across all permutations of the features joining the model. SHAP explains individual predictions and the results can be visualised in heatmaps that reveal what groups of pixels or superpixels had the most importance in a prediction, providing insight into which vehicle or background features had the most weight in the detection decision. Formally, SHAP is computed by S. Lundberg 2018,

$$E[f(X)|X_s = x_s]$$

where f is explained model, and S a subset of features X of fed to the model

2.6.2 Layerwise Relevance Propagation

Layerwise Relevance Propagation (LRP) was introduced in Lapuschkin et al. 2015 and was developed especially to explain deep learning models. In comparison to SHAP values which forward inputs to the model, LRP focuses on the backpropagation of the model predictions. LRP is based on the principle that the total relevance is conserved when back propagating it from layer to layer, where relevance is defined as the activation strength of an output node for a certain class Böhle et al. 2019. Different LRP variations exist, each with different protocols for passing relevance from one layer to another, and the version used in this study is the $\beta - rule$. This rule is chosen in Böhle et al. 2019 because it allows for the tuning of the weights given to the position contribution and negative contribution to the decision of the model. The LRP results can be visualised in a heatmap that portrays the input space in which each voxel contribution's importance to the final decision is represented. The method provided accurate results.

2.6.3 Grad-CAM

Grad-CAM or Gradient-weighted Class Activation Mapping was introduced in Selvaraju et al. 2019 and initially developed for explaining decisions of convolutional neural networks. It is common practice to the target layer as the last convolutional layer of the network, where the target layer is the unique network layer that the Grad-CAM explains. This layer choice is motivated by the fact convolutional layers naturally retain spatial information, and because the last layer building a higher-level feature space is the best compromise between high-level semantics and detailed spatial. Grad-CAM is calculated by first backpropagating the gradients for a target class c to the chosen target layer k, obtaining neuron importance weights, α_k^c representing "a partial linearisation of the deep network downstream from feature map activations A_k" Selvaraju et al. 2019. Then, Grad-CAM is calculated as follows:

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha_k^c A^k)$$

Experiments

3.1 Transfer Learning

The aim of this experiment is to assess the effectiveness of Transfer Learning in the task of damage detection on vehicle images. To do so, four datasets described in 2.1 are used: NRB, NEOKT, Stanford Cars, and Roboflow. The NEKOT dataset was initially created for a classification task between damaged and whole vehicles. The classification dataset is artificially converted to an object detection dataset by keeping the manually labelled class but synthetically adding bounding boxes labels localising car objects in the images using a Yolov5 model pretrained on the Microsoft Common Objects in context, or COCO, dataset Lin et al. 2014 that contains a class 'car'. It must then be noted that the model performances might be biased towards detecting 'whole' cars since the bounding boxes are generated on 'car' objects. Examples of this labelling can be found in A.1. All four datasets are partitioned into 400 validation images, and 99 testing images, and the remaining images are used for training.

For all conducted experiments, the Yolov5 model was trained using the default data augmentation techniques, and the default hyperparameters, suggested in Jocher et al. 2022, except from batch size that was set to 64, image size to 416, and the maximum number of epochs set to 50 except for fine-tuning where it is set to 100. The default augmentation techniques are composed of an exhaustive augmentation list. Namely, rotations, sheers, gain scaling, horizontal flip, and colour space scaling such as HSV saturation augmentation, random cropping, blur, and grayscale conversion. Moreover, the images are shaped into mosaics, which are basically random superpositions of the augmented images.

The model's performance was evaluated using the metrics of mAP on 0.5 IoU threshold, precision, and recall. To evaluate the effectiveness of Transfer Learning, two approaches are used: training the model by freezing the backbone layers (the 10 first layers), and training the model without freezing these layers with a twice smaller learning rate. All Yolov5 models are initialised with the COCO dataset pretrained weight. Meaning that for some experiments, the model would have been given three different trainings. First, an underlying training with the pretrained COCO weights, then a transfer learning and fine-tuning training. Further, a specific learning rate schedule was used for every experiment which is the default one suggested in Jocher et al. 2022 except for fine-tuning where the initial learning rate halved. The default learning rate schedule is composed of a warm-up phase, where the learning rate is increased linearly from 0 to the initial learning rate, and a decay phase, where the learning rate is decreased exponentially, from the initial learning rate to the final learning rate which is set by default to 0.01. The warm-up phase is used to prevent the model from getting stuck in a local minimum, and the decay phase is used to prevent the model from overshooting the minimum and to allow the model to converge to a better solution. Then, even though a maximum number of epochs was set, the training is set to stop early if the model's loss on the validation set does not improve for 15 epochs. This early stopping is done deliberately to avoid overfitting. In addition, the learning rate schedule design also further prevents overfitting since it reduces the learning rate gradually as the training progresses, which correlates with the overfitting risk of the model.

The results of these experiments will provide insights into the effectiveness of Transfer Learning in the task of detecting damages on vehicle images and will help to identify the best approaches for achieving optimal performance in this task. The transfer learning experiments are designed to assess the transfer of knowledge between different tasks and datasets. By comparing the performance of the model on different transfer learning scenarios, we can understand the impact of Transfer Learning on the model's performance and determine the optimal way to utilize this technique in the task of damage detection on vehicle images.

Transfer from Stanford Cars to NRB Here, the model is first trained to recognize cars and then to recognize their damages. This may present a challenge due to the visual dissimilarity between the car and damage objects, as a car may be characterized by features such as four wheels, four doors, and a steering wheel, while damage may be represented by irregular lines, and broken glass. These two kinds of features seem visually distant from each other.

Transfer from Roboflow to NRB Here, Transfer Learning involves the transfer of knowledge related to damage detection, although the class labels differ between the two datasets. Roboflow contains more diverse and accurate damage types labels but lacks labels informing on the damage location, in contrast to NRB. This experiment will evaluate the effectiveness of Transfer Learning in transferring labels.

Transfer from NEOKT to NRB Here, Transfer Learning involves the detection of both damaged and undamaged cars, followed by the specialized task of detecting damages on cars. This training procedure may be particularly relevant as the model is first trained on a more general task before being specialized for a more specific one.

3.2 Image Generation

3.2.1 GANs

The architecture proposed in Antipov, Baccouche, and Dugelay 2017 is utilised to generate images of undamaged vehicles. The first step in implementing this model is to train a GAN capable of generating vehicle images from a Gaussian latent space. The next challenge is to train an encoder used to approximate the latent vector of an input face image while preserving the identity of the input image.

As explained in 2.4.1, GANs can be laborious to train. Thus, the following hyperparameter tuning variations are considered in order to find the optimal Generator/ Discriminator training equilibrium.

- "batch size": choice([64,128])

- "latent dimension size": uniform(50,500)
- "learning rate": log uniform(0.0001,0.0005),
- "Loss Function": choice(["DualContrastive"Yu et al. 2021,"Wasserstein" Arjovsky, Chintala, and Bottou 2017,"BCE" Goodfellow et al. 2014a,"Hinge" Lim and Ye 2017])
- "Grandient Penalty Strength": choice([0,5,10]) a Regularisation of the magnitude of Discriminator gradients Gulrajani et al. 2017.

Introducing some randomness in training :

- "target value flip probability" : uniform(0,0.5)
- "target value smoothing probability" : uniform(0,0.5)
- "random weights initialisation": choice([True,False])
- "generator dropout" : choice([True,False]) Adding dropout layers between the building blocks of the Generator
- "discriminator dropout" : choice([True,False]) Adding dropout layers between the building blocks of the Discriminator

To find the best combination of hyperparameters, the tuning python library Tune Liaw et al. 2018 is used with the Pytorch Lightning python library Falcon and team 2019. The GANs are evaluated by evaluating their generated samples using the FID metric, introduced in 2.5.1, implemented in the Torchmetrics Python implementation Detlefsen et al. 2022. The FID metric is used to compute the distance between the two image group distributions. A sufficient number of generated images to accurately characterise their distribution. In the original FID paper Heusel et al. 2017, 50,000 images are used to compute an FID score. Due to limited computational resources, only 1000 generate images are involved in the FID computation procedure. Further, the optimisation procedure involes the training of 100 different GAN samples in order to explore the GAN optimality space and find a local or global minimum. The approach taken to sample from the hyperparameters space is to use a population-based training (PBT), a scheduling algorithm introduced in Jaderberg et al. 2017 and implemented in Liaw et al. 2018. PBT is a method inspired by a genetical algorithm, where each member of the population (known as a "worker") can draw information from the rest of the population during the training, to improve their own performance. Further, PBT can be customised with the following parameters: the time attribute is set to 'training iteration', the perturbation interval is set to 3, and the burn-in period, or the period in which no perturbations are applied, is set to 15.

3.2.2 Stable Diffusion

Stable Diffusion offers the ability to guide the image generation process using a positive and a negative prompt, setting positive and negative weight towards and against the prompts. The following will be used, '*A picture of a car*' as positive and '*a picture of a damaged and broken car with broken glass and scratches*' as a negative prompt. The model also includes the ability to adjust the weighting of these prompts through the use of a "guidance scale" parameter. Additionally, the model's generative power can be controlled through the use of a "strength" parameter, which determines the number of diffusion steps taken and therefore the extent to which the initial image features are preserved. The strength is a value set from 0 to 1 where 0 is that the model has no freedom to modify an image and 1 is it has total freedom and with a full diffusion process, the initial image features are totally lost.

The implementation of the Stable Diffusion model is carried out using the Python diffusers python library Rombach, Blattmann, Lorenz, Esser, and Björn Ommer 2022. The models' performance is evaluated using SSIM, VRS, and VSS metrics introduced in 2.5. A baseline experiment is first conducted to determine the range of values for the strength and guidance scale parameters. A sample of images is generated from a grid-search-based setting at [0.2-0.7] strengths and [7-50] guidance scales. Then, an experiment is conducted to compare the ability of the SSIM and VSS metrics to assess the preservation of the identity of a vehicle in the generated image. Using the most relevant metric, further experimentation is conducted to find the optimal combination of guidance and strength parameters for generating undamaged vehicle images. The optimisation is performed using the Optuna Python library Akiba et al. 2019. Once the optimal generation parameters have been determined, their effectiveness in improving the damage detection capabilities of Yolov5 is tested by incorporating the generated image pairs into the training dataset. The undamaged generated images are juxtaposed with their original damaged version. Two different labelling versions of the augmented dataset are investigated, either with or without labels indicating the location of the original damage on the generated images.

3.3 Explainability

In this experiment, the decision-making process of the model for damage detection and classification is explored. The effectiveness of three explainability methods 2.6 is evaluated by comparing the heatmap visualizations they produce. The focus is on identifying, assessing and comparing the regions of the input image that have the greatest influence on the model's output, comparing the results of each method, for each class. To ensure consistency in the experimental results, all experiments are conducted using Yolov5 damage detections on images of shape (416,416).

SHAP values analysis Here, the analysis is conducted using the SHAP python library S. M. Lundberg and Lee 2017. First super-pixels are defined as a discrete grid-shape partition of the image. The width of the super-pixels is set to 52, resulting in a total of 64 super-pixels per image. It is worth noting that increasing the number of super-pixels significantly increases computation complexity.

LRP Here, the analysis is conducted using yolo-heatmaps python library Karasmanoglou, Antonakakis, and Zervakis 2022. The LRP analysis is run in contrastive mode, meaning that positive and negative contributions towards the detection/classification are both considered. The weights set on the contributions are b1 = 1.0 for the positive contributions and b2 = 0.5 for the negative contributions.

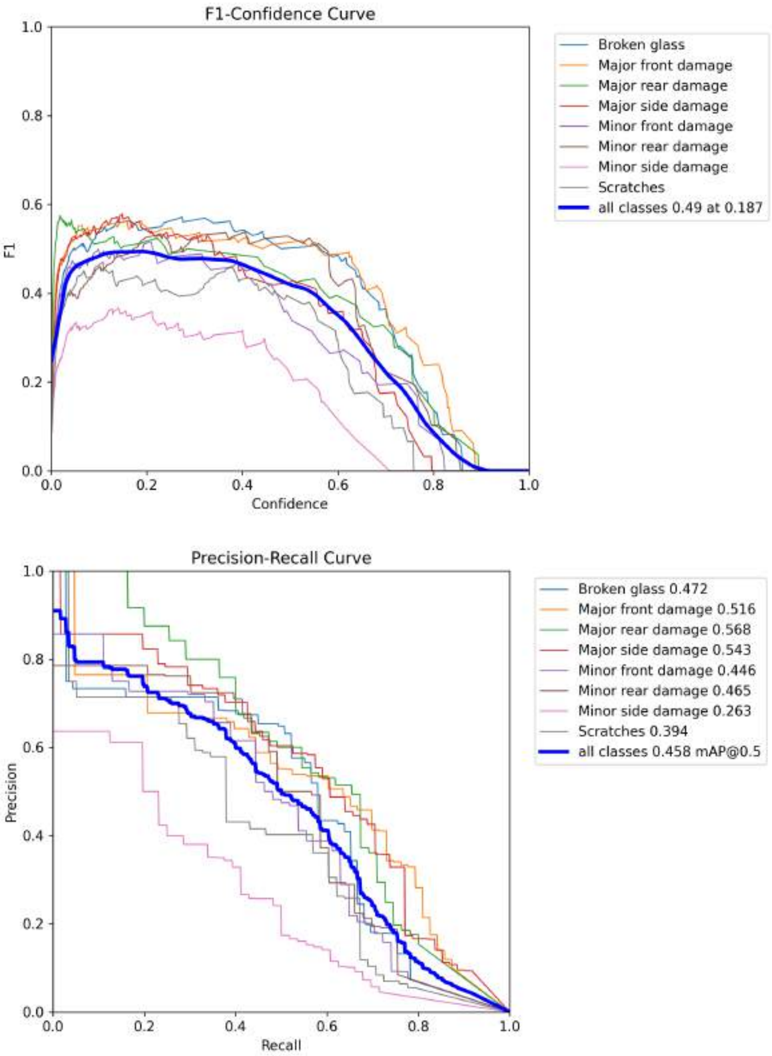
Grad-CAM Here, the yolov5-gradcam Python Library is used Mohammadi Kazaj 2021. The last and third convolutional layer of the first to last model module is investigated, the last module being the classification and detection Head and the layer before being the layer that captures the highest level feature representation captured by the model.

Results

4.1 Transfer Learning

In this experiment, the performance of several transfer learning and fine-tuning approaches from the different considered vehicle datasets to the NRB dataset are compared. Figure 4.2 shows their training logs. Namely, purple - Stanford Cars, pink - COCO, blue - NEOKT, yellow - Roboflow. All models were pretrained on their respective datasets for 50 epochs before being transferred to the NRB dataset. The last black curve represents the training logs of the baseline model, trained for 50 epochs on the NRB dataset without specific transfer learning or fine-tuning procedures. It is observed in 4.2 that none of the transfer-learning and fine-tuning models achieved equivalent precision, recall, and mAP performances as the baseline model (in black). Moreover, it is also observed that models pretrained on the Roboflow and Neokt datasets, which are more specific and closer to the NRB detection task, performed worse than the models pretrained on the more general Stanford Cars and COCO datasets. Then, a performance drop is observed between the transfer learning and fine-tuning training logs which are respectively the left and right parts of the curves in 4.2. As a reminder, at this dropping point, all the model layers are unfrozen, and the learning rate is increased and halved. Finally, it is noted that the pink model is essentially the black model, but trained with the same transfer learning and fine tuning training procedure as the other models. With this in mind, it is observed that the purple curve, representing the Stanford Cars pretrained model, is superior to the black curve.

The baseline Yolov5 detection model is formally evaluated using a testing set composed of 400 instances, containing at least 50 instances of each damage class. The first part of the evaluation is shown in Figure 4.3 in which the investigation is done using the metrics precision, recall (b), and F1-score (a) over the different damage classes. The blue curves are averaging the metrics among all damage classes. From those averages, it is found in (a) that the optimal confidence value is at 0.187 where the F1 rises to 0.49. Derived from the F1 score definition, this value represents the optimal balance between precision and recall for the baseline model. Further, it is found in (b) that the mAP of the baseline model is 0.458, corresponding to the area under the averaging curve. Regarding the comparison between the detection performances of the different classes, it is found that major damages are better detected than average, whereas scratches and minor damages are detected worse than average. This can be seen in the F1 and Precision-Recall curves which are respectively higher, and lower, than the averaging curves.



(a)

(b)

Figure 4.3: (a) F1-Confidence Curve, and (b) Precision-Recall curve, assessing the baseline Yolov5 damage detector

The confusion matrix for the optimal baseline model on the testing set, shown in Figure 4.4, revealed that most parts of the classification errors occurred between similar severity or location classes such as minor rear and major rear, or major side and major front. Further, scratches and minor side damages were the classes that most frequently confused the model. They can be seen as similar severity classes. Lastly, the Broken glass class has almost not been confused with any other classes. Figure 4.5 shows an aggregation of the confusion matrix described in 4.4 on the different severity and location types, respectively. Looking the theses confusion matrices generalisations, it can be observed that overall the class types' confusions are fairly low, <10%. Moreover, a confusion of 9% is observed between the front and the side damage and a confusion of 8% is observed between Minor and Major damages.

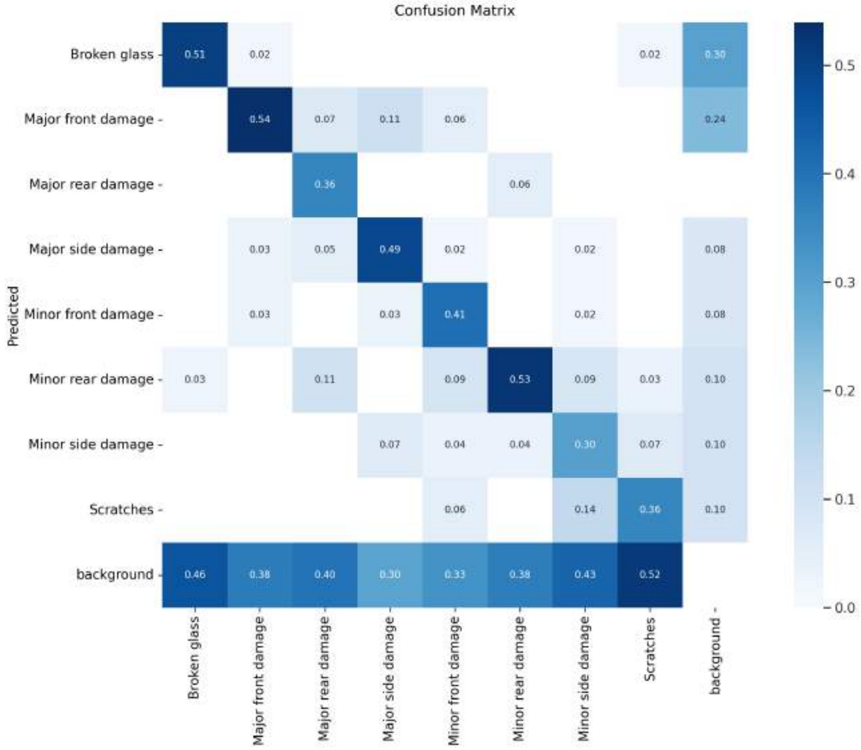


Figure 4.4: Confusion matrix of the different damage classes, assessing the baseline Yolov5 damage detector, on optimal confidence level, 0.187, found in 4.3

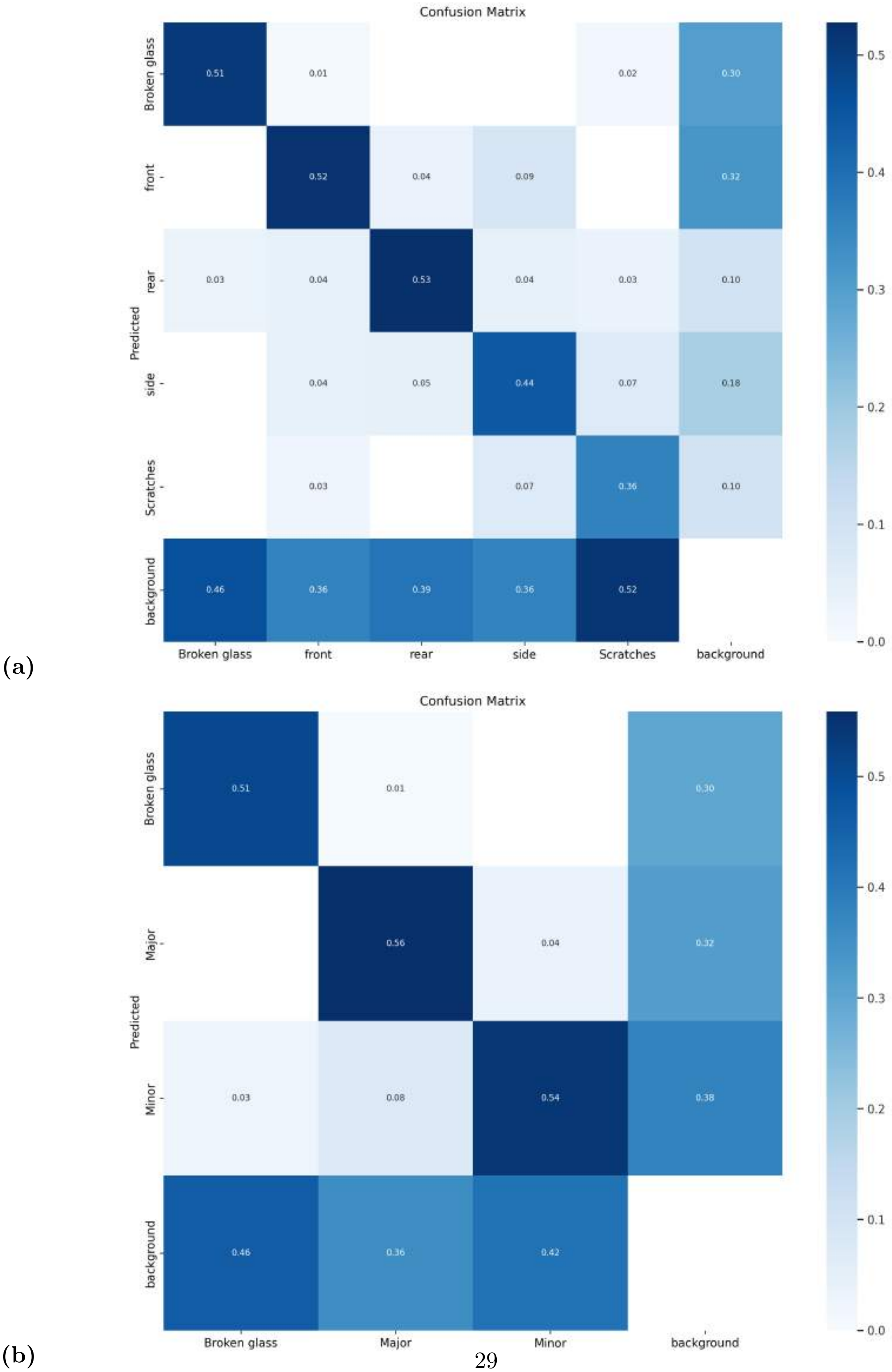
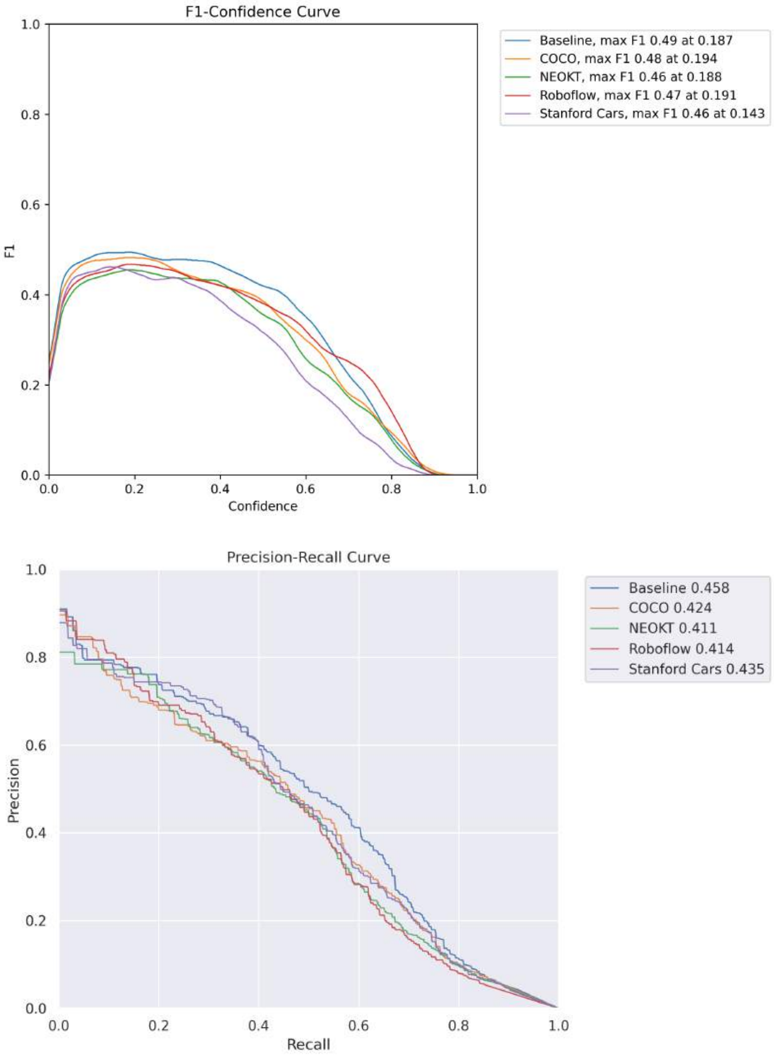


Figure 4.5: Aggregation of confusion matrix 4.4 on (a), damage locations and (b) damage securities, with scratches considered as minor

Figure 4.6 offers a more detailed comparison between the different fine-tuned models and the baseline model on F1 and precision-recall curves. It can be observed in (a) that the Roboflow pretrained model overtakes the baseline models for a small region around 0.8 confidence, and in (b) that the Stanford Cars pretrained model overtakes the baseline model on a small region around 0.3 recall and the Roboflow model around 0.1 recall. The comparison is further detailed by exploring the mAP differences in the different damage classes. The results are shown in the appendix A.2. It can be observed that the Stanford Cars pretrained model is the best-performing model on all major damage classes. However, the model suffers from a low mAP, (-12%) in the Scratches class. In contrast to the Stanford Cars pretrained model, the major damage classes major side and rear performed worse in the Roboflow pretrained model, (-10%), and the Broken glass and Major front damage classes increased in mAP by 2% and 5%, respectively. Linking back to the damage classes contained in the Roboflow dataset, crack and hole, medium deformation, severe deformation, severe scratch, slight deformation, slight scratch, and windshield damage. It can Broken glass is the only class that is not contained in the Roboflow dataset in the form of 'windshield damage'. It can also be observed that the Roboflow classes do not contain information on the damage location, hence the damage classes are specialised in damage severity and generic type assessment.



(a)

(b)

Figure 4.6: Comparison of testing curves of the fine-tuned models and the baseline model on (a), F1 curve and (b) Precision-Recall curve

Table 4.1 summarises the results found in 4.6 and provides a comparison with the model developed by Bruyn 2022. The IoU threshold used in Bruyn 2022 is not mentioned thus it is assumed that the default IoU threshold of 0.6 was used. Using the same train, validation and testing sets are the ones used in Bruyn 2022, a 44.7% mAP was observed on the baseline model, which is 10% higher than the mAP of 34.9% achieved by Bruyn 2022. Moreover, a 49% F1 score was observed on the baseline model, which is 14% higher than the F1 score of 35.9% achieved by Bruyn 2022.

Model	Baseline(COCO), IoU=0.5	COCO	NEOKT	Roboflow	Stanford Cars	Baseline, IoU=0.6	Bruyn 2022
mAP	45.8%	42.4%	41.1%	41.4%	43.5%	44.7	34.9%
max F1	49%	48%	46%	47%	46%	49%	35.9%
at confidence	18.7%	19.4%	18.8%	19.1%	14.3%	19.2%	45%

Table 4.1: Results of the Transfer Learning and Fine-Tuning experiments, in comparison to the Baseline and results of Bruyn 2022

4.2 Image Generation

4.2.1 GANs

As mentioned in 3.2.1. The first step in the acGAN image-generation pipeline is to get equipped with a decent generative model before tuning it on identity preservation. The training was done using a set of real images of the Stanford Cars dataset. It was observed that damage images often portray close-ups of the vehicle, thus, a random centre cropping augmentation technique is applied with a random cropping size limited to two-thirds of the initial size, followed by a random rotation of up to 5 degrees. 4.7 shows a sample of the real images used to train the GAN.

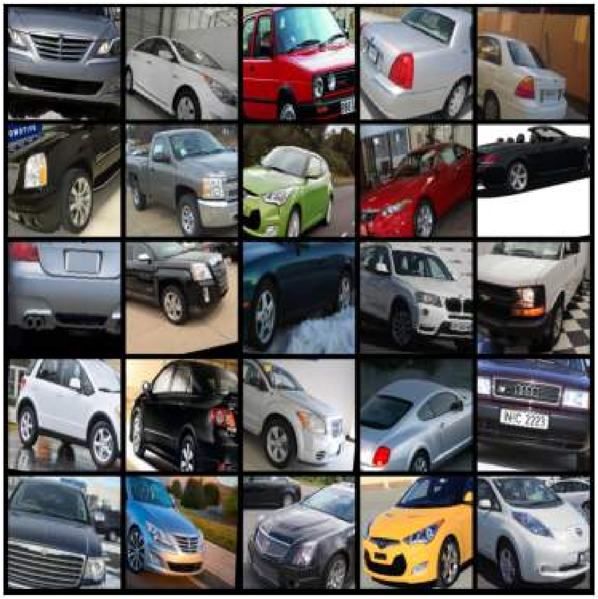


Figure 4.7: a batch sample of Stanford Cars augmented images

Then, the baseline Generator-Discriminator modes are trained using the following hyperparameters.

- batch size: 64,
- latent dimension size: 100,
- learning rate: 0.0001,
- Loss function : Binary Cross Entropy,
- probability flip: 0.0,
- target value smoothing : 0.1,
- input dimension: 64,
- generator dropout : 0.1,
- discriminator dropout : 0.1,
- Gradient Penalty strength: 0.0,
- weights clipping value:0.01,

After 330 epochs, the real score, the score granted by the discriminator to the real images rises to 1 and fake score, the score granted by the discriminator to the generated images, drops to 0, indicating that the discriminator have become too performant and have overtaken the Generator. This resulted in the training process becoming frozen, as the generator was unable to improve the quality of the generated images. However, figure 4.8 shows that the generated images still resemble real vehicle images.

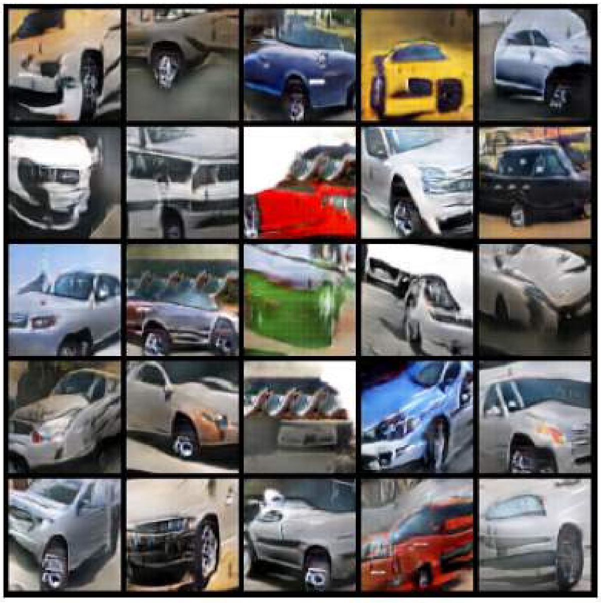


Figure 4.8: 25 samples of images generated with a GAN, after 330 training epochs

The second planned experiment on GAN fine-tuning, described in 3.2.1, did not lead to usable results due to technical and image quality evaluation challenges. The training process is computationally intensive, requiring one week of computations for training 100 models on 10 epochs with the following GPU processor: GP100GL [Tesla P100 PCIe 16GB], 64bits. Additionally, image generation evaluation was a burden as described in 3.2.1, a large number of generated images are needed to obtain a representative FID score for the generated image distribution. This evaluation step slows down the training procedure. Furthermore, it is unclear whether FID is an effective metric for identifying the best or potential GANs. Figure 4.9 compares the best GANs model selected according to (a) FID and (b) my own subjective evaluation. It can be questionable whether FID is a relevant metric for guiding the GAN fine-tuning process towards better quality GANs. Finally, while the main goal of the study is to develop a framework towards a vehicle damage assessor, the image generation is just a subset of this goal, and the GANs training process optimisation was found to be too time-consuming to be included in a study, which also explores transfer learning, explainability, and Stable Diffusion, which showed more promising results than GANs. Nevertheless, training and fine-tuning a GAN for this specific task remains a worthwhile research topic that could be explored in future studies.

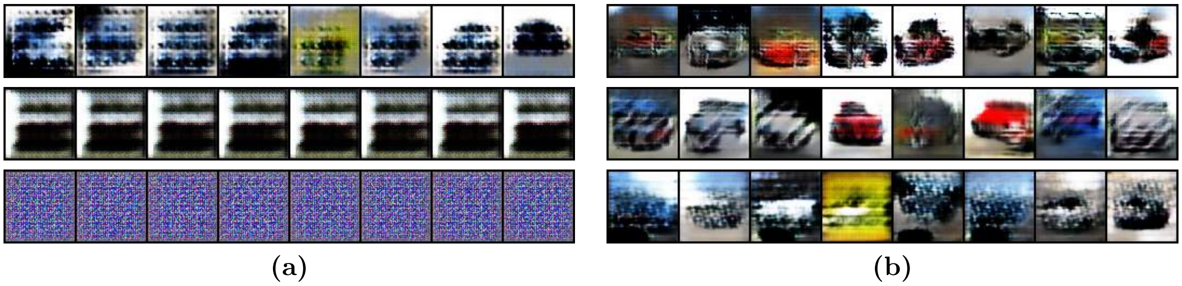


Figure 4.9: the best sample of GAN generated image selected by, (a) FID, (b) myself

4.2.2 Stable Diffusion

Generative Model Optimisation

4.10 shows the results of the first baseline experiment, generating samples of generated pairs using different parameters for the model. It is observed that the model is able to generate images that are more easily distinguishable as vehicles compared to those generated by GANs (4.8. Especially at low strength, the parts of the image which are modified the most are the parts where the damage is localised. At high strength where the model has the most generative power, the generated cars resemble either old fancy, sports or futuristics cars, at the bottom left. It is to be noted that the model's generative performance is also influenced by the severity of the damage. Another sample of generated images is shown in 4.15 where the damage is much lighter than in 4.10 and the initial characteristics of the car are better preserved.

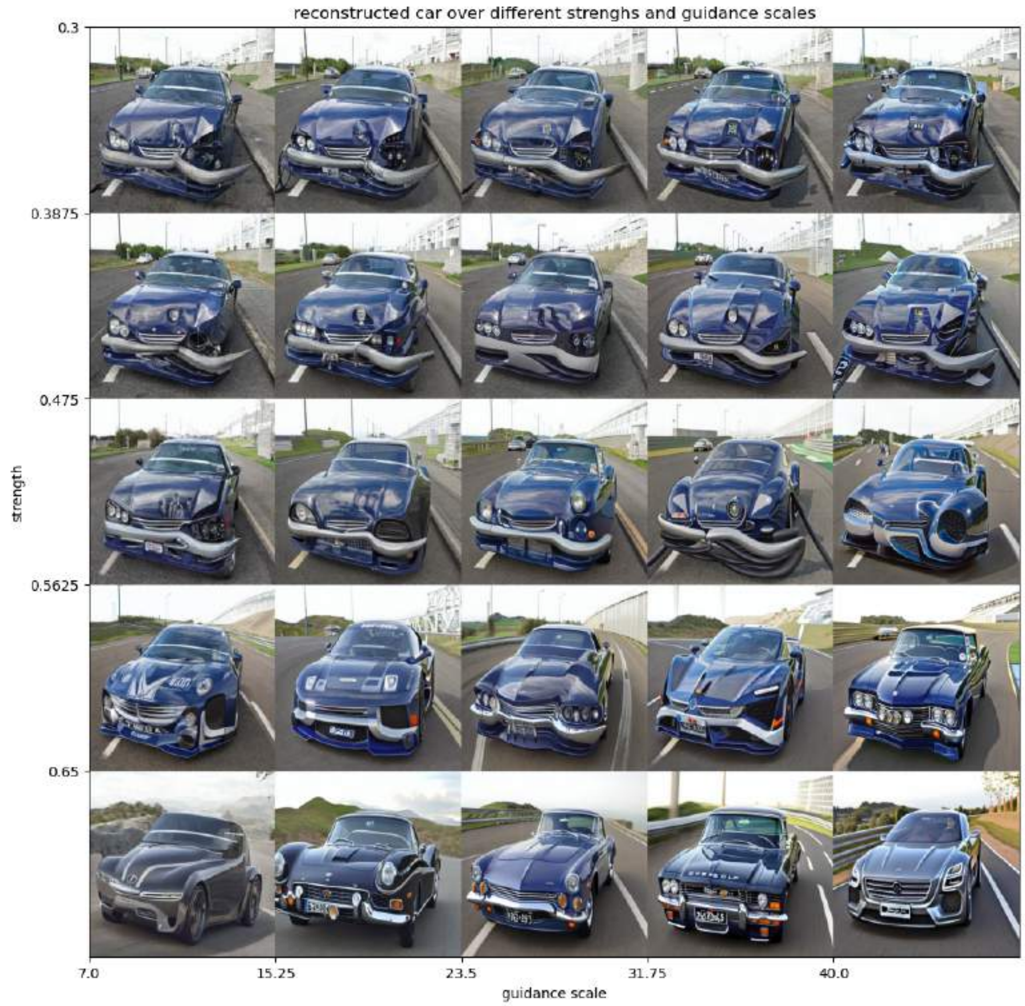


Figure 4.10: 25 samples of synthetic images generated with a diffusion model at different strengths and guidance scales

Then, 4.11 shows the generated images introduced in 4.10 evaluated and sorted by SSIM and VSS. We can first compare the evaluations of SSIM and VSS since they are evaluated on a similar purpose. It is first observed that the range of metrics $(1.37 - 0.73)/0.73 = 0.88$ and $(0.61 - 0.31)/0.31 = 0.96$ is 8% higher for SSIM. It is observed that the sorting of the images is quite similar from one metric to another. In fact, there is a correlation of -81% between the metrics values. It is also noted that VSS has placed the futuristic car in the last position, while SSIM has placed the Mercedes-type car with the orange background.

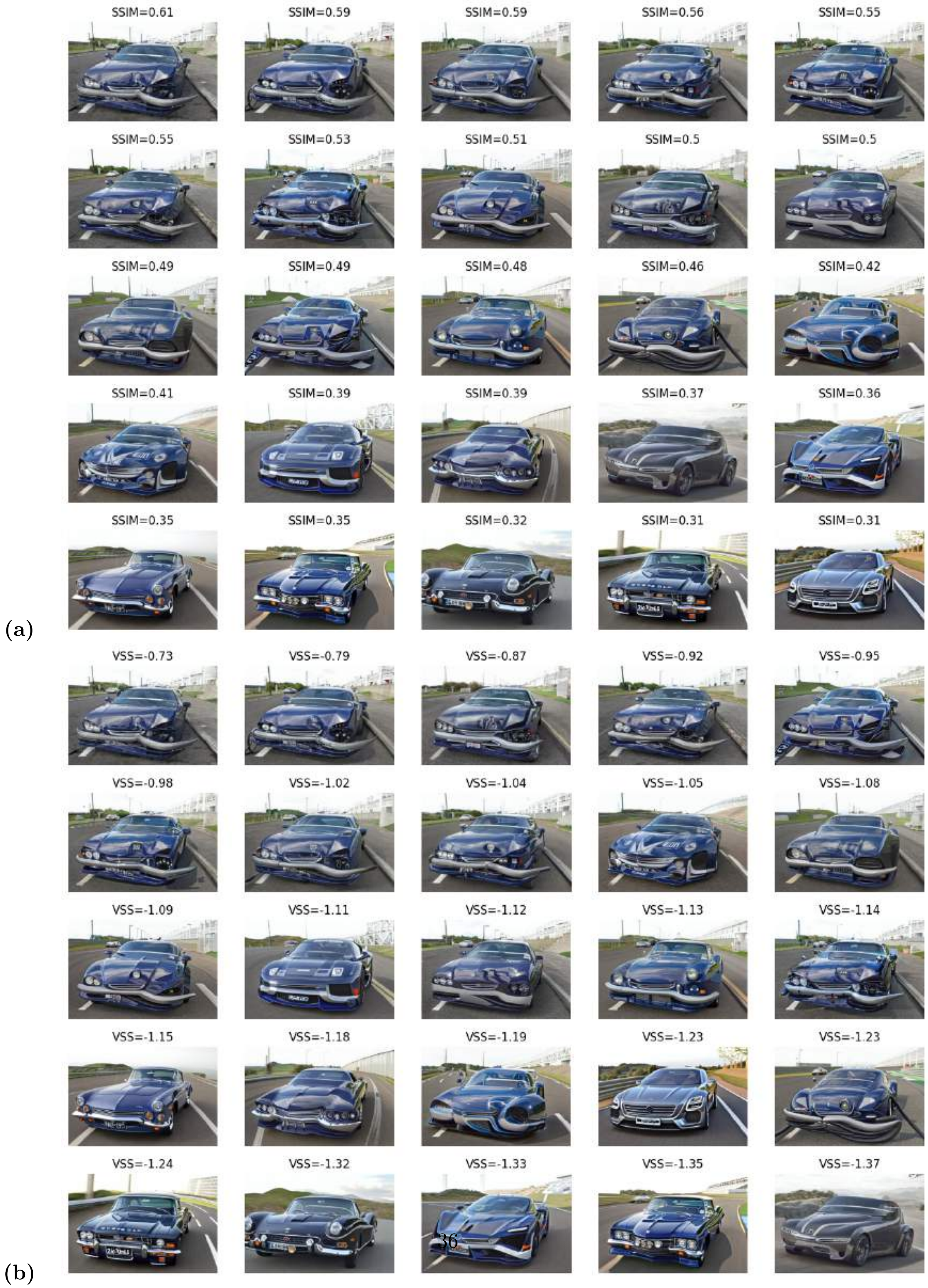


Figure 4.11: a batch sample synthetic images generated with a diffusion model, evaluated and sorted with, (a) Structural Similarity Index Metric, (b) Vehicle Similarity Score

Figure 4.12 illustrates the VSS performances on comparing vehicle images. The top left image is compared to the rest. It can be observed that the first four images all portray cars of sedan/coupe type cars which are similar to the compared image, while the last images portray SUVs, Vans and sports cars.

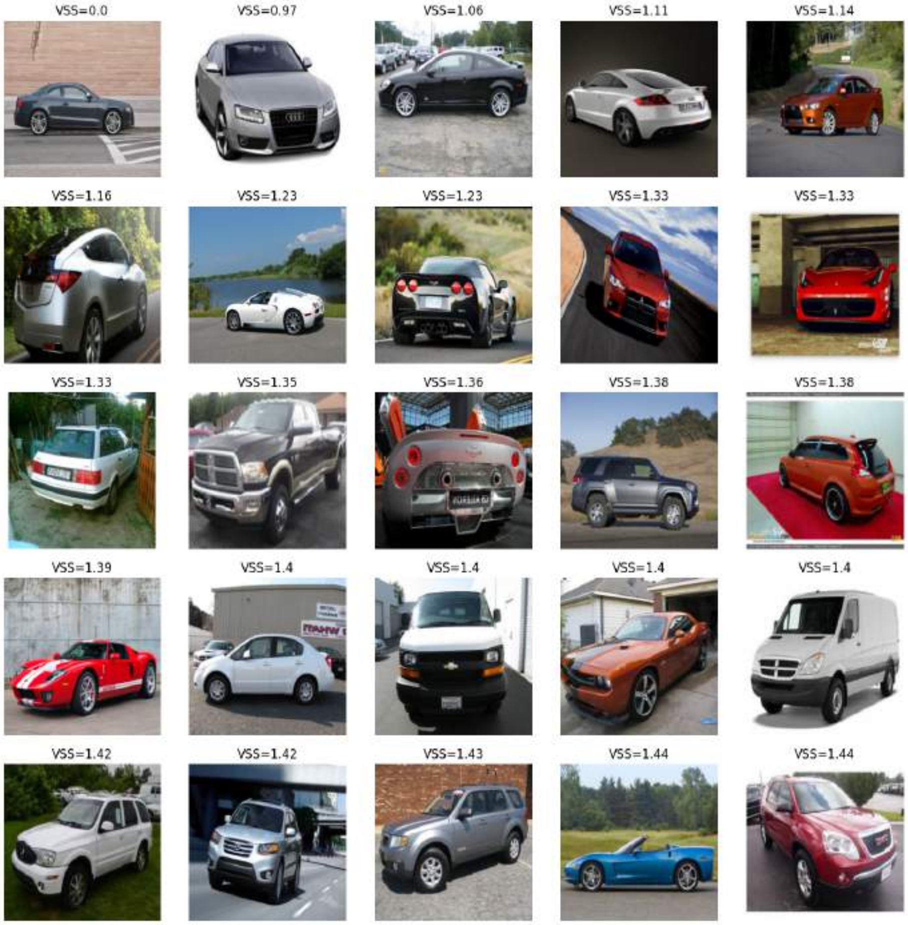


Figure 4.12: a batch sample Stanford Cars dataset, evaluated and sorted with, Vehicle Similarity Score, in comparison to the top left image

Figure 4.13 shows generated images evaluated by VRS. It shows that the images with the highest scores are the ones that portray better-looking vehicles. It is also computed that the VRS and SSIM metrics values given the images are correlated by 0.90% and with VSS by -0.80%.

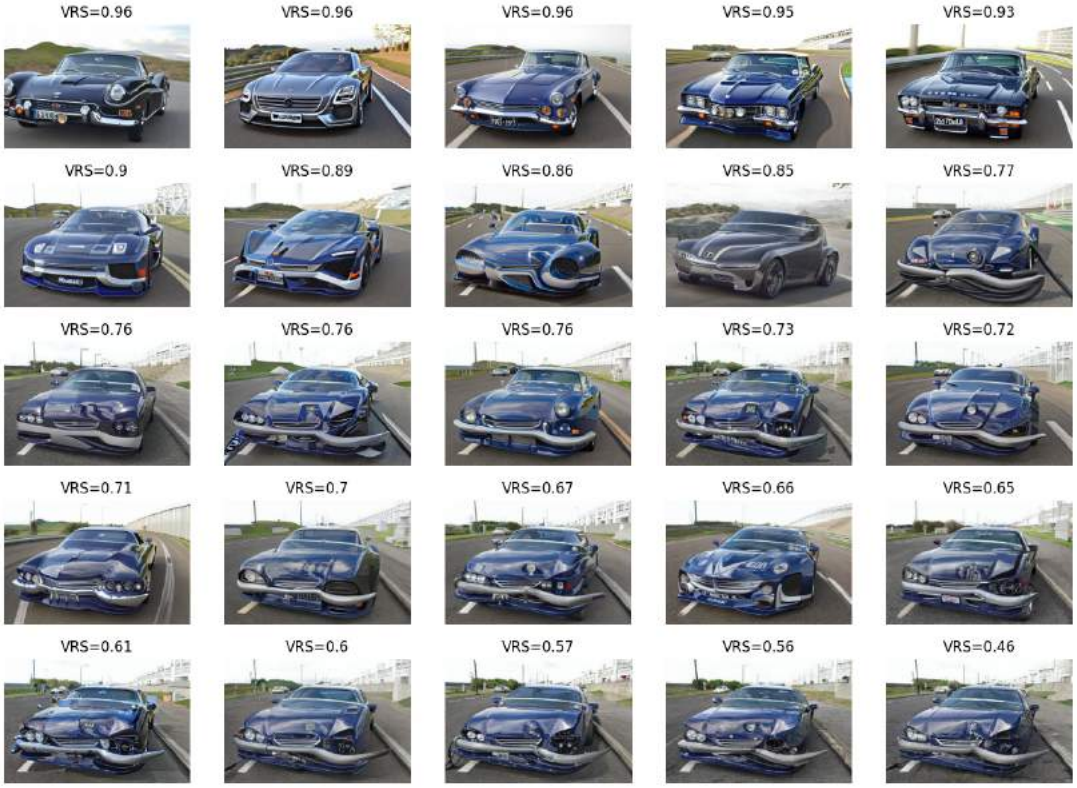


Figure 4.13: a batch sample synthetic images evaluated and sorted with Vehicle Resemblance Score

For the reasons discussed in 5, VSS is chosen as the most relevant metric for evaluating the preservation of the identity of a vehicle in the generated images. The final image metric used for image evaluation is then given by $Score = VSS + VRS$. The next experimentation is conducted to find the optimal combination of guidance and strength parameters for generating undamaged vehicle images, using the $Score$ metric. Figure 4.14 portrays a heatmap of the model evaluations at different strength-guidance pairs. A model is evaluated by taking the average $Score$ of a sample of 25 images of generated images. 4.14 reveals an optimality region located approximately between 0.52-0.69 strength and 35-45 guidance scale. However, it is unclear whether there is a global optimum. Nevertheless, the optimal value observed within the 50 trials is strength: 0.67 and guidance: 43.07.

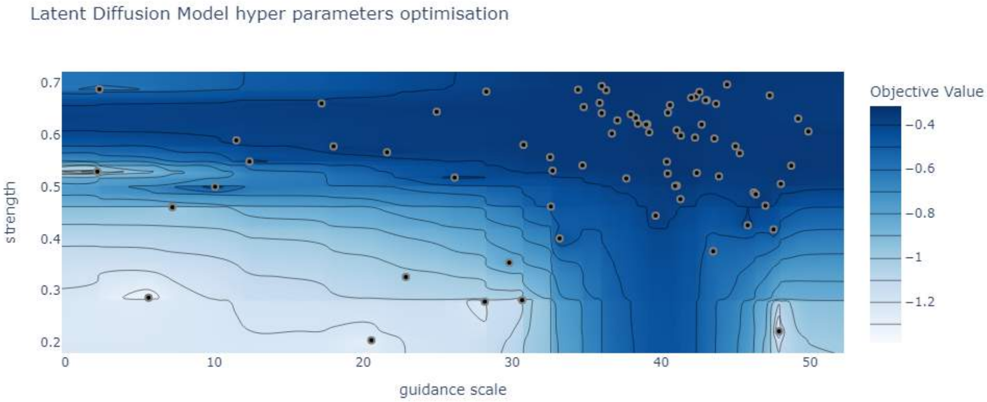


Figure 4.14: The guidance scale and strength hyperparameters of the stable diffusion model are tuned in order to select the pair that generates the best undamaged vehicles

Using the optimal values found in 4.14, the NRB dataset is doubled by selecting the highest-scoring image out of a sample of 5 generated images. This is illustrated in 4.15, where the original vehicle damage image is shown on the top left and the selected image, with the highest score (0.94) among the other 4 generated images, is shown to its right.

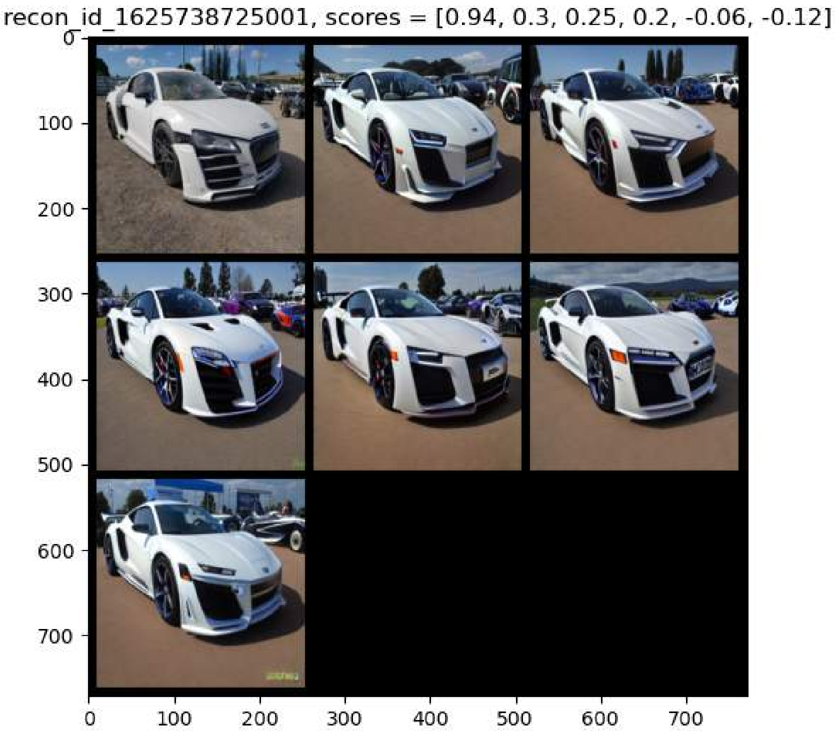

recon_id_1625738725001, scores = [0.94, 0.3, 0.25, 0.2, -0.06, -0.12]
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700

Figure 4.15: 5 samples of synthetic images sorted by (Vehicle Resemblance Score - Vehicle Similarity Score), generated from the top left image

However, it is found when further exploring the generations on the test set, shown in A.3, that even though the model seems not intimidated in building reconstructions of heavily damaged cars, it has been found to have trouble with images portraying close-ups of the vehicles, often found in minor damages and scratches where only the damaged part of the car is showed. Those lead to the model generating images that could be recognised as an entire vehicle instead of its close-up.

Including Synthetic data in the Detection Model

Two different variations of the synthetically augmented dataset are experimented with, illustrated in 4.16.

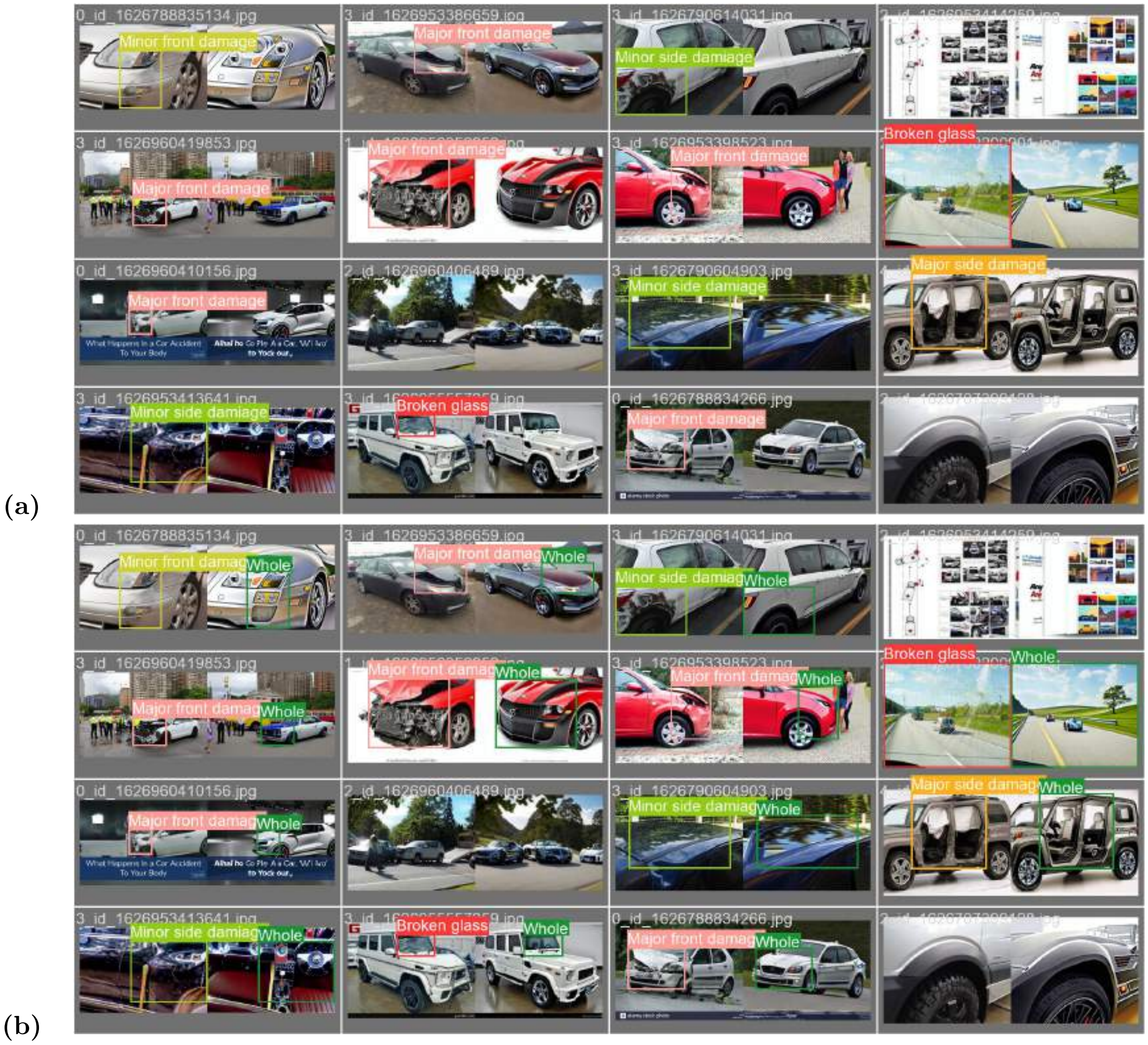
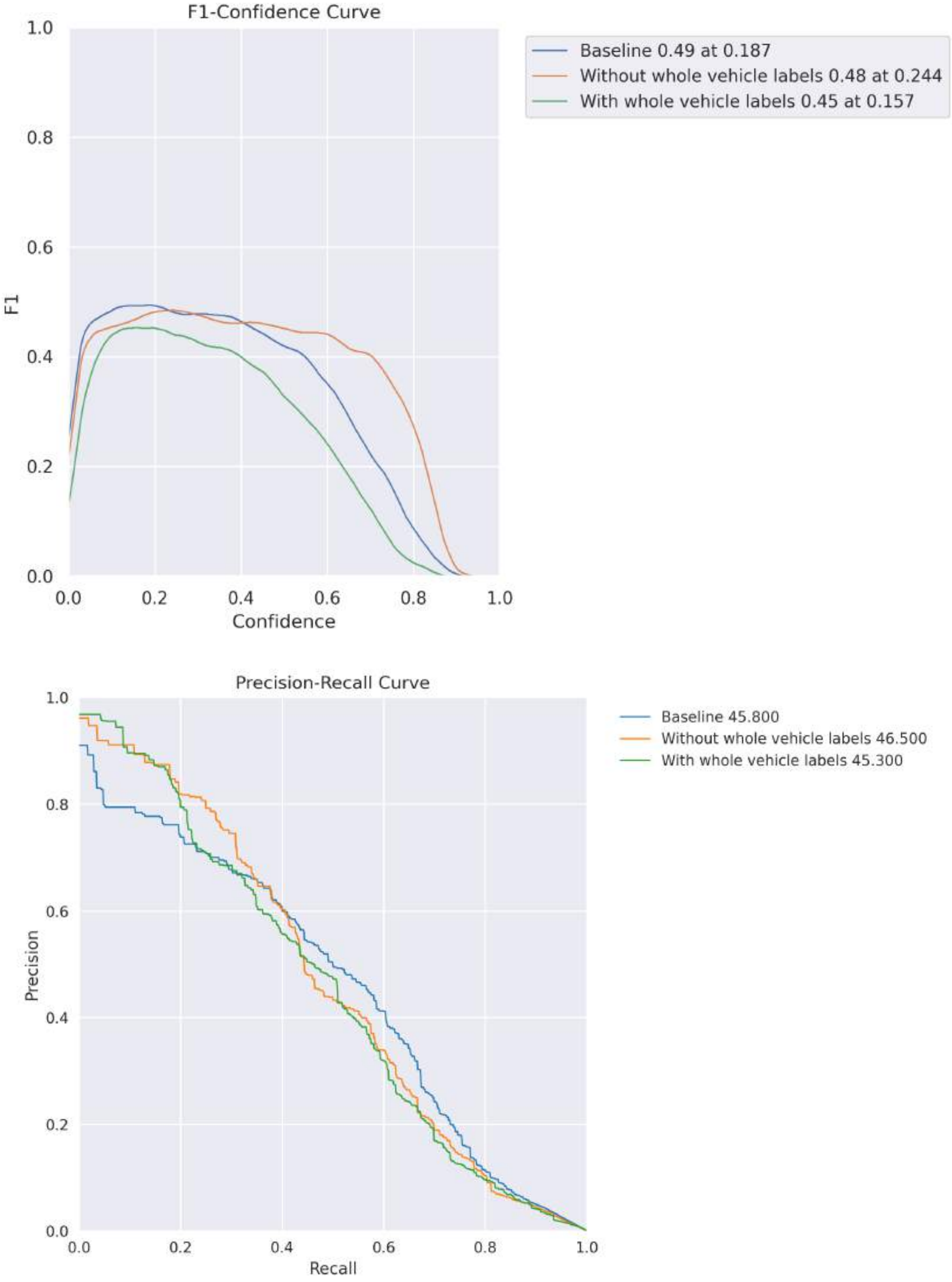


Figure 4.16: (a): a batch sample of the synthetically augmented NRB dataset, (b): (a) with an additional 'whole' class label

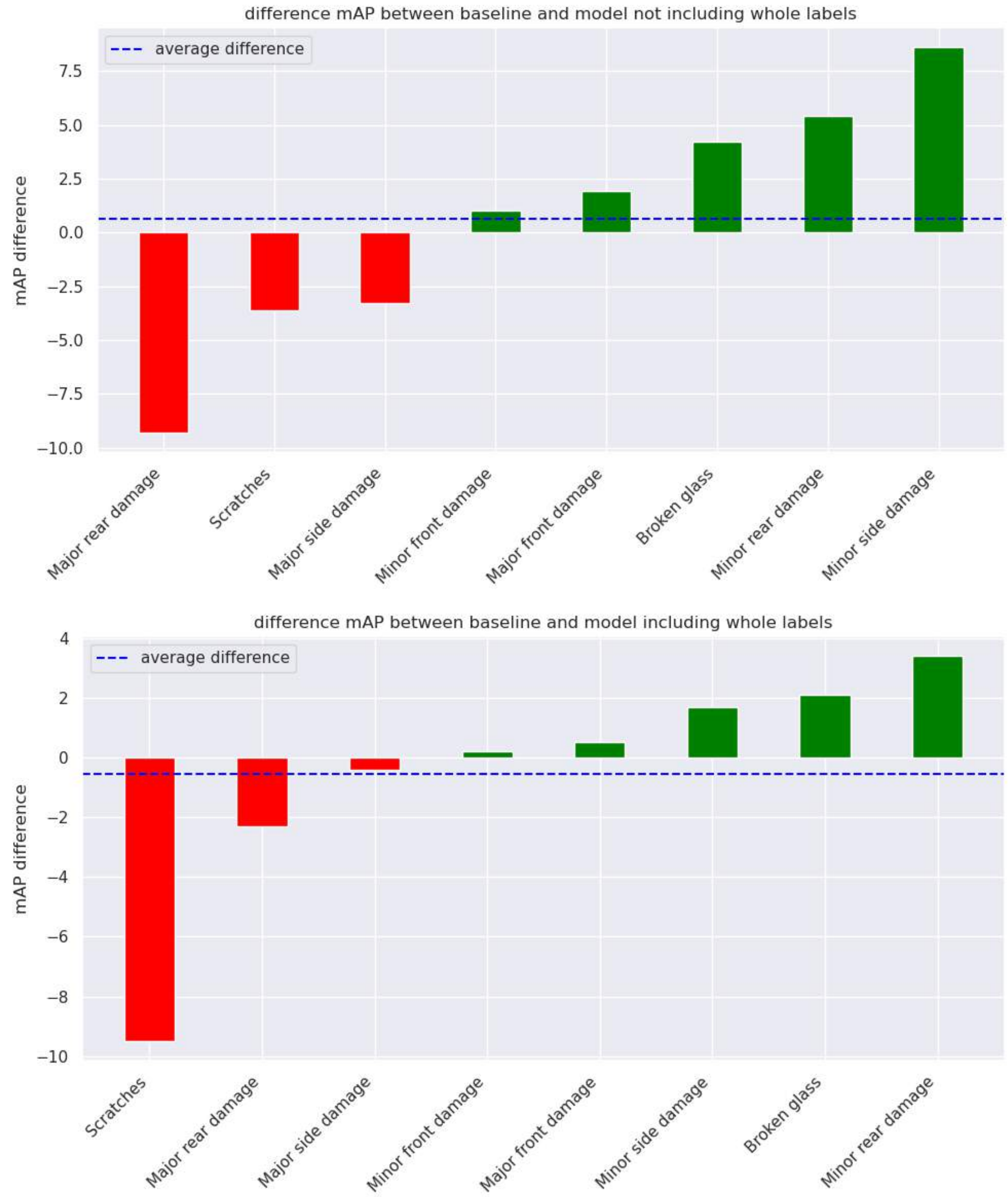
It is observed in 4.17(a) that even though the baseline model is 1% F1 higher, half of its F1 curve lies below the model trained without 'whole' labels. Moreover, the mAP results, shown in 4.17(b), revealed that the model trained without whole labels had a slight mAP improvement of less than 1%. This is further illustrated in 4.18, where the mAP differences per damage class are shown. It is observed that both models trained with and without whole labels had a better performance in the Broken glass and Minor side and rear damage class, while worse mAP performances are observed for the Scratches and the Major side and rear damage classes.



(a)

(b)

Figure 4.17: Comparison of testing curves of models trained on synthetically augmented datasets and the baseline model on (a), F1 curve and (b) Precision-Recall curve



(a)

(b)

Figure 4.18: Damage class-specific differences in test mAPs between the baseline model, and the models trained (a) excluding a 'whole' label, see 4.16(a), and (b) including a 'whole' label, see 4.16(b)

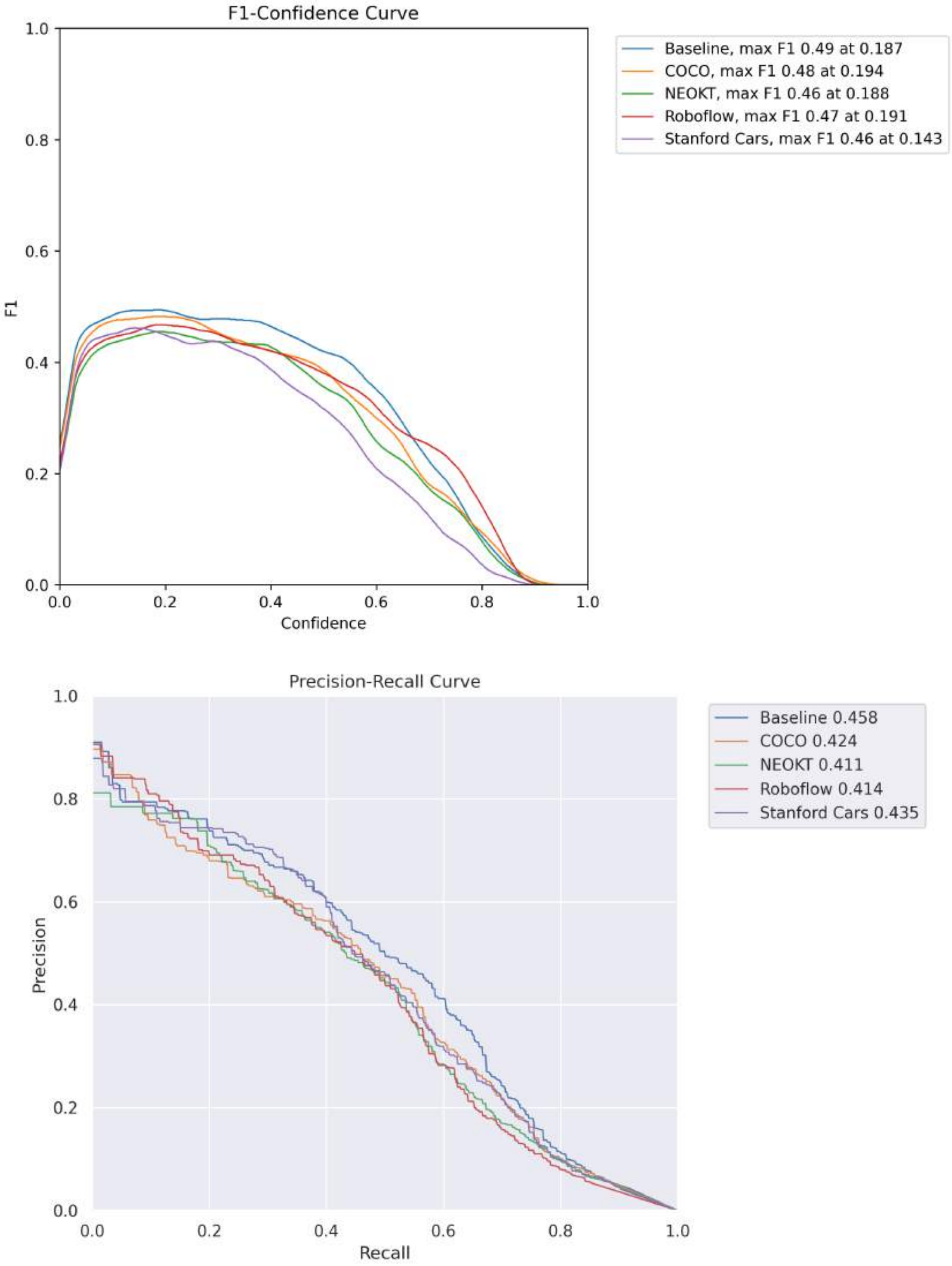
4.3 Explainability

In this experiment, the model decision-making process on damage detection was investigated using three explainability methods: SHAP values analysis, LRP and Grad-CAM. The experiments aim at exploring the regions of the images that led to the correct or wrong model predictions, and compare the results of each method for each class. Except for Figure 4.19, the results of the explainability experiments are included in A.5 since they take up a lot of space. Figure 4.19 offers 5 different visualisations of the same image. From left to right, the first image shows the damaged car with the bounding box localising the damage on the image, followed by the same image but with the damage bounding boxes generated by the model instead. In this case, the model has detected a major side damage, which corresponds to the true label. Next to that image, the LRP contrastive explanation is shown on a yellow background. The red regions had a positive contribution to the detection while the blue ones had a negative contribution. Then, the Grad-CAM explanation is shown with a coloured frame bounding the detected damage. A heatmap is included in this frame where warm colours indicate a high model response to that region of the image while cold colours indicate a low or null model response. In the last image, is shown the SHAP visualisation of the image where dark red colours indicate a high superpixel influence in the final model decision.

The results of the experiments demonstrated that all three methods were able to highlight the regions of the input image that have the greatest influence on the model's output. However, when comparing the methods, each had its strengths and weaknesses. The LRP is found to be the most precise, making it more useful for accurate explanations. The SHAP values analysis is found to be less precise, even though the precision can be increased by reducing the size of the superpixels, but at great complexity costs. However, in comparison to the other methods, SHAP has the advantage of providing information on the background around the detection, whereas LRP and Grad-CAM focused on the bounding boxe regions. Grad-CAM, on the other hand, was found less useful since it is less precise than LRP and provides no information on the background. It can be considered as a summary of the other two methods, highlighting the center(s) of attention of the model. For some cases, it may be useful to have a broader explainability view, especially when the explainability response of the other methods is too complex, making the visualization appear cluttered.

The results of the experiment revealed specific regions of the input images that were highlighted for different damage classes. For the broken glass class, the holes or cracks in the windows were identified as significant regions A.13, A.14. In other cases, the model focused on the interior of the car to infer the presence of broken glass, or in some cases, the past presence of glass A.15, that is now absent. For the class of major front damage, the bumpiness at the back of the vehicles A.25, some opening of the hood of the car A.27, or the front lights of the vehicle A.26 were identified as significant regions. Regarding the major rear damages, it was also found that the model paid attention to the rear bumpiness A.24, and major defects in the vehicle body A.28. However, it was also observed that the model did not pay much attention to the front of the vehicle or especially the vehicle mirrors, which could provide clues about the position of the vehicle and differentiate between front and rear damages. Further, on minor damages, it was found that many of the images featured close-ups of the damage and the model paid attention to specific regions such as broken lights A.31, small bumps on the hood A.33, and the entire front of the vehicle for minor front damage A.34. Regarding minor rear damages, the highlighted regions were the vehicle back bumper A.35, the backlights A.36, or another frequent rear damage, shaped like a small round bump A.38. The model also frequently focused on the back corners of the vehicle A.40, A.39. For minor side damage, the door edges A.43, gas tank door A.44, or the wheel region A.47, were highlighted. Comparatively, major side damages highlighted larger regions of the vehicles, such as the vehicle doors A.59 and the wheel and front fender areas A.60, or both A.58. Finally, the highlighted regions for scratch damages were the actual scratches on the body of the vehicle A.57.

The previous paragraph presented the explanations for true positive labels, when the model's predictions were correct. This section delves deeper into the analysis of false positives, or incorrect decisions made by the detection model. One first example, shown in A.17, illustrates a situation where a rectangular shape in the damage has attracted the attention of the model and might have led to the major front prediction instead of rear. In another example, A.18, the model also predicted major front damage instead of rear damage but this case is hard to evaluate, even for humans. The model could have used the mirror position information to deduce the position of the car. Further, some borderline examples were observed where the damage lies between the front and the side of the vehicle as shown in A.19, and in A.21. Other borderline examples are observed between the scratches class and other minor damage classes, as shown in A.54 and A.55. Furthermore, it was observed that some wrong decisions may be caused by questionable labelling as shown in A.20, A.37, A.52, and A.32, A.45. It is also noted that is sometimes even hard for humans to provide a label due to the image's bad quality, as shown in A.22, or due to the zoom on the damage being extreme, as shown in A.46, A.41, A.45 providing little contextual information on the damage position on the vehicle. Then, the limitations of the model's ability to distinguish between actual damage and light or shadows reflections on the vehicle body were observed in A.56, where a linear light reflection was mistaken for a scratch, and in A.23, where a damage was incorrectly detected on a black spot of the body of the car that is in reality a shadow reflection. Additionally, some other detection errors were found to be due to image artefacts, as seen in A.49 and A.51. Furthermore, some examples of general model failure were observed in A.50 where the front radiator was incorrectly identified as scratches, and in A.30, where even though the detection was correct, the model's attention was not directed towards the actual damage in the image. Other failures included the model missing a round bumper in A.53, focusing too much on the front of the car while the damage was a side damage in A.29, and mistaking a side damage as a rear in A.42 and A.48. Interestingly, the model has identified a child as a damage in A.16, which is left as food for thought.



(a)

(b)

Figure 4.1: Comparison on the testing set of the different pretrained models in comparison to the baseline model on (a), F1 curve and (b) Precision-Recall curve

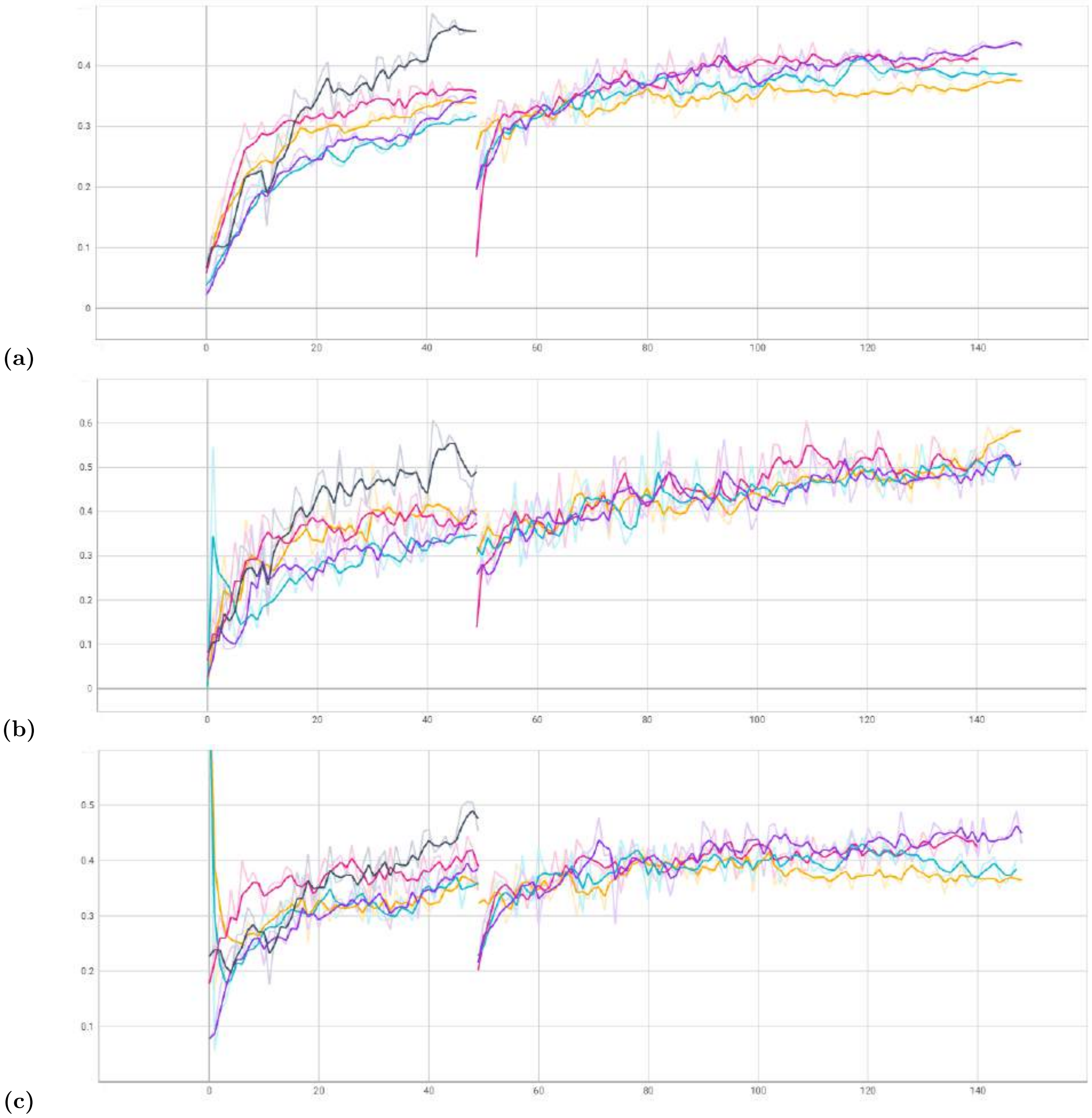


Figure 4.2: Validation metrics over the different training epochs of the transfer learned models, left part of the logs, and their fine-tunings, right parts of the logs, (a) mAP (IoU > 0.5), (b) precision, (c) recall, of the Baseline damage detection in black, Stanford Cars in purple, COCO in pink, NEOKT in blue, Roboflow yellow.

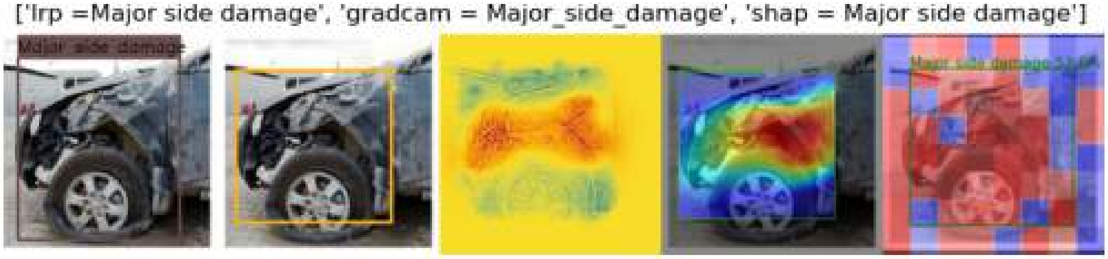

['lrp =Major side damage', 'gradcam = Major_side_damage', 'shap = Major side damage']
Major_side_damage
Major side damage

Figure 4.19: minor side damage

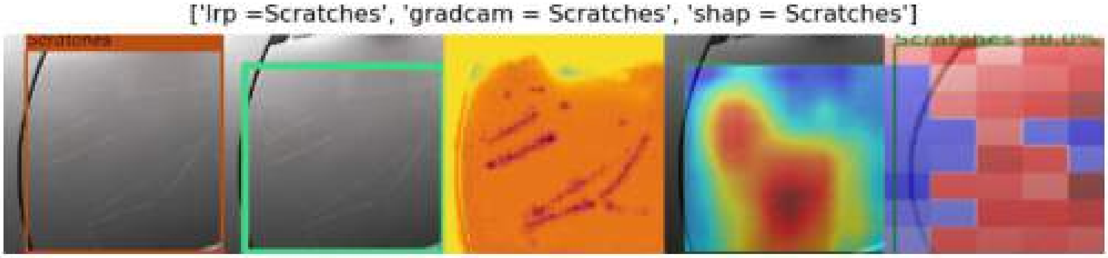

['lrp =Scratches', 'gradcam = Scratches', 'shap = Scratches']
Scratches

Figure 4.20: scratches

Discussion

On the question, **Does Transfer Learning help improving the model's performances?**, the transfer learning and fine-tuning experiments showed that none of the models achieved comparable mAP, precision, and recall performance compared to the baseline model, which was trained solely on the NRB dataset and which achieved 48.5% mAP and 49% F1. Vis-a-vis existing research, the baseline model also outperformed the model trained in Bruyn 2022. This performance discrepancy may be attributed to a different data augmentation technique used, creating a more robust model, a different learning rate schedule used to optimise the model training or a combination of these factors. Then, it should be noted that the comparison of this model performance to other models from the literature is limited in relevance, as it has been seen that the model performance is influenced by the choice of the class labels as well as the data imbalance. For example, Fouad et al. 2022 reached 85% F1 but on a classification task but used less confusing classes, namely, broken head light, tail light, broken glass, scratches, and dents. However, Bnspriyatham 2022 was only able to reach 79.4% F1 on damage localisation classification, and 68.8% for severity classification task. On a damage detection task, Ruitenbeek and Bhulai 2022 achieved a mAP of 41,3% by training Yolov3 on 2500 web scraped images of bend, crack, dent, glass shatter, light broken, missing, rust, scratch classes.

The datasets considered in the experiments contain tasks lying somewhere between the COCO detection task of common objects and NRB detection task of damages. The experiments thus suggested that transfer learning is more effective when applied from a general task to a specific one. This is further illustrated by models pretrained on more specific datasets, Roboflow and NEOKT, performed even worse than those pretrained on more general datasets, Stanford Cars and COCO. It can be explained by the fact that models pretrained on more general datasets may contain more generic features that are useful for a broader range of tasks, including the NRB damage detection task and that it may be easier to convert from generic to specific features in 'forward' manner rather than converting from specific to specific features, where the pretrained model has been trained on an overlapping task and optimised into a global or local minimum that may not be a suitable starting position for reaching the local or global minimum of the transferred task. It may be worth further expanding the investigation by transferring a pretrained model that hasn't reached optimality. The big drop observed between transfer learning and finetuning training curves may be explained by the fact that unfreezing the topmost layers breaks some training equilibrium and the model takes a new optimisation path. Further, it was observed that the stanford cars pretrained model performed better than the coco pretrained model, which had the same training procedure. It is thus believed that there is a good chance that a Stanford cars pretrained model, when trained with the same training procedure as the baseline model, might surpass the latter. Observing the differences between the baseline mode and the model pretrained on Stanford cars, the lower mAP (-12%) on the scratches class might mean that scratches might require features that are far away from features used to train whole cars, developed in the Stanford cars pretraining.

The evaluation of the baseline model using the testing set revealed satisfactory performance, as evidenced by the high values of mAP, and F1-score. However, the confusion matrix 4.4 highlighted that the model struggled to distinguish between similar severity or location classes. The observed confusion between the scratches and minor side damages could be due to the fact that scratches commonly occur on the side of the vehicle. Moreover, the confusion observed between the front and the side damage may be caused by damage located at the boundary of those two classes, especially when the vehicles are heavily damaged. Furthermore, some confusion was also observed between Minor and Major damages in the same location, especially the rear. This may be due to the subjectivity of terms such as "minor" and "major" damage, which were labelled by a human and may not align with some 'computable' logic, thus expose to human errors, also discussed by Ruitenbeek and Bhulai 2022 which suggested implementing cross-validation between multiple annotators to meet this shortcoming. Nevertheless, the class 'Broken glass' is found to be an adequate label, since it has been observed that it has almost not confused with any of the other classes. The observations of the model's performance with respect to different damage types suggest that minor damages are more challenging to detect and decrease the model's overall performance. This could be due to minor damages that could be confused with scratches, and false positives might be generated on light and shadow reflections on the vehicle's body that resemble damage. This phenomenon is consistent with findings in previous literature **shadow22**, and Ruitenbeek and Bhulai 2022 which suggested reflection removal filters such as polarising filters.

Further improvement may be attained through the acquisition of more varied and representative training data, as well as the design of labels that are less subjective and less likely to lead to confusion for the model. Additionally, the incorporation of a class label for reflections might be useful to actively train the model to detect those and to differentiate them from the other damages, which might reduce false positive detections. Finally, to ensure the validity of the results of the transfer learning experiments, it would be more prudent to conduct a cross-validation procedure on them.

On the research question, **How to generate an undamaged version of a damaged vehicle image?**, a GAN model with identity preservation was considered but ultimately not utilized due to challenges encountered during its training. Instead, a Diffusion Model guided by a positive and a negative prompt and using the image of the damaged vehicle as the initial image was utilised and evaluated. The results of the GAN experiments suggest that while GANs are capable of generating images that resemble real vehicle images, the GANs trained in this study are not yet able to generate high-quality images that can be used to train a vehicle damage detector. It has been decided to discard the GAN improvement research path due to the challenges that rose from their training and because diffusion models showed to be an effective candidate. However, future research may further improve the GANs model by fine-tuning the training hyperparameters, testing other GAN architectures from the wide diversity of architectures that exists, and identifying more effective image quality evaluation metrics.

The results of the Stable Diffusion experiment showed that the model was able to generate high-quality images showing reconstructions of undamaged vehicles. However, it was observed that the model struggled to accurately reconstruct vehicles with scattered silhouettes resulting from severe damage, especially when low generative power was granted. Despite this limitation, the model demonstrated the ability to generate coherent reconstructions of heavily damaged vehicles. This could be interpreted as the model being able to abstract the damages present on the vehicle and reconstruct a vehicle resembling a whole vehicle, using the information available on the damaged vehicle image.

It was also found that the model's performance was influenced by both the severity of the damage and the chosen parameter values for which an optimal region was identified. However, it was observed that the model controllability and predictability were limited, especially when given large generative power since it generated vehicle reconstructions from various qualities. The issue was addressed by anyway setting the model to the optimal combination of guidance and strength parameters found, and by generating a set of 5 images in hopes that at least one of them accurately represents the undamaged vehicle. Another potential solution to limit the model unpredictability that could be explored in future studies is to investigate the model's sensitivity to different, more specific and constraining image prompts, as only a single general prompt was used in this study.

The results of the generative model also revealed a limitation in its ability to accurately reconstruct close-up images of damages, such as those typically associated with minor damages and broken windows. The model was found to generate an incoherent reconstruction of the damage, by extrapolating the image to guess vehicles in the structures of the image. This shortcoming is believed to be due to the model's guided to focus on generating whole vehicles, and not recognising any vehicle close-ups due to their distance from whole vehicles. The model then tries to extrapolate and anyway fit a vehicle in the given input image. This shortcoming might be addressed in future studies by using a set of specialised models for reconstructing different close-ups distances of the vehicle, or by again tuning the guiding prompts, stating that the pictures might be pictures of an entire vehicle or close-ups of it. Another observed limitation of the model diffusion is that it has the tendency to generate either old, recent, or futuristic fancy or sports cars. This bias might be introduced with the training dataset of the model, over-representing those types of cars.

The effectiveness of using the generated images to improve the damage detection capabilities of the baseline model was tested by incorporating them into the training dataset. The results showed that the use of the generated images did not lead to significant improvement in damage detection performance compared to the baseline model. However, diving down into the results, it is found that the decrease in performance was limited, with a 1% reduction in average F1 and a 1% increase in mAP. The analysis of the mAP differences per class revealed that the scratches class had a significant negative impact on the model's performance. As it has been observed that vehicle reconstructions can be defective, generating malformed reconstructed vehicles, it is interpreted that the detection model may be confused with those malformations, which might look like scratches or other types of damage. Future research could be conducted to assess this hypothesis by placing more weight on the vehicle resemblance score in VSS when selecting the best model and reconstructions.

Further, on the research question, **Which metrics, best evaluates and separates an undamaged vehicle from a damaged vehicle?**, the Vehicle Similarity Score and the Vehicle Resemblance Score metrics were introduced. They were specifically designed for the image generation application of the study. The experiments showed that they were able to quantify the similarity of two vehicle images and to evaluate the general resemblance of an image to a vehicle, respectively. The metrics were combined by subtracting the resemblance to the similarity, and the combination could be further tuned by adding some weights on the individual metrics. The VSS metric was also compared to another notable metric for image comparison, SSIM. Despite the large similarity in image evaluations provided by the two metrics, showed in **??**, VSS is empirically preferred over SSIM since it is less likely to be attracted by artefacts such change in background colors, whereas VSS's evaluation has been shown to be more driven by the shape and brand aspects of a vehicle 4.12.

The experiments conducted in this study provide insight into the effectiveness of GANs and diffusion models in generating images of undamaged vehicles. However, the results also indicated that there is still room for improvement in the performance of these models, as well as in the evaluation and guidance of the diffusion model.

On the last research question, **To what extent can we explain the decisions of a damaged vehicle image detector?**, the study limited its investigation to the model detections and a future study could extend to explaining the reasons behind the absence of detections. The results generally revealed that the model's attention was drawn to regions of the image corresponding to the class of damage being detected, with specific attention regions identified for each class. The examination of the model's incorrect decisions revealed certain patterns in the model's defective behaviour and some of them are discussed here. For instance, in instances where the glass was so broken that it was no longer visible, the model may have inferred that the detection of an interior of a vehicle indicated a lack of glass. Then, in an instance where a rectangular shape illustrated in the explainability visualisations may have attracted the attention of the model and thus led to the major front prediction instead of rear, it can be interpreted that this rectangular shape was identified as a front light by the model, which is a landmark for front damages. Furthermore, in the instance where the model could have used the mirror position information to deduce the position of the car, it is noted that no lights were visible, resulting in limited landmarks for the model to differentiate front from rear damage. This type of scenario might pose a significant challenge for the model. Then, in an instance, the model's attention was not directed towards the actual damage in the image, and instead focused on the presence of the light, which might infer a risk that the model assumed that the image contains damages. Lastly, in an instance where the model missed a round bumper, it is believed to be due to overflowing with scratches, introducing noise and fooling the model. Lastly, the model was found to be missing a round bumper, which is believed to be due to overflowing with scratches, introducing noise, and fooling the model.

It was found to be sometimes unclear how the model made the distinction between major and minor damages, and front and rear, for which the confusion was highlighted in the confusion matrix 4.4 of the model. However, it was found that the model uses the back bumpers to make its decisions on rear damages, and it is believed that the model utilises features such as the shape of the front and rear lights to make such distinctions. Furthermore, no investigation was done in explaining the model decisions of images that lack damage detection, which is left to a future investigation. Then, the results of the synthetic data augmentation did not provide satisfactory results and it is suggested to further investigate of how the model utilizes synthetic data is necessary to understand its effects on the model's decisions. This explainability study provided insight into the limitations of the current approach, and those suggested avenues for further research to improve the model's ability to explain its decisions and its performance in detecting damaged vehicles. The task of classifying damages was found to be non-trivial due to the large inter-class similarity of damages and ambiguity in labelling, which for some severity and location borderline cases complicated the classification process for location and size of damages. This was also observed by Ruitenbeek and Bhulai 2022. Thus, the suggestion of Bruyn 2022 to use some more disparate damage types that could be more logically defined and quantified such as broken headlight or damaged tire, is restated in this study. Furthermore, it is believed that the wide variety of damage shapes and types made the task of localizing damage non-trivial. It was also noted that close-up images limited the context information available for the model and thus limited its performance. Furthermore, it was found that Grad-CAM might be less useful than the two other methods since the last and third convolutional layer of the first to last model module capturing the highest level feature representation provided redundant and less precise explanations. However, Grad-CAM might be exploited differently by exploring visualisation maps of different layers of the model. This leaves a lot of information to explore and decrypt, and it may be found class-specific detection features with low certitude, but at a great amount of work costs.

The study has highlighted numerous areas for future research. Firstly, there is room for improvement in the performance of GANs for generating undamaged vehicle images and in the evaluation and guidance of diffusion models. Additionally, further investigation may be needed to determine the effectiveness of transfer learning for vehicle damage detection, including the use of different training procedures, a cross-validation procedure and the transfer of models that have not yet reached optimality. Furthermore, it is needed to explore the use of more specific and constraining image prompts to limit the unpredictability of the model, as well as the potential use of specialized models for close-up damage images. Then, the study suggests further investigation into the subjectivity and overlapping of damage labelling and the development of more objective metrics for evaluating and separating undamaged from damaged vehicles. Also, more research is needed to explain the decisions made by a damaged vehicle image detector, including the investigation of missing detections and the examination of other layers within the model. One potential cheap way to improve the model's performance is to upgrade it to the latest Yolo release which is theoretically better than the previous versions. Investigating the use of fully synthetically generated datasets, with generation guided on the different classes, might also be interesting research, some examples of those hypothetical datasets are shown in A.4. Further, this whole study might not have been necessary if some better dataset existed, the field is missing some benchmark datasets. The dataset quality improvement can come from solely augmenting the number of images, which was observed by Dwivedi et al. 2019 "we observed that Transfer learning performs best already available large dataset". It can also come from the focus on limiting the number of image contexts and varying the number of different representations of the instances, as stated by Dwivedi et al. 2019 that "collection of a more diverse dataset would be sufficient to implement an automated vehicle damage identification system in the near future".

Conclusion

This study aimed at improving the performance of a single-stage object detection model, Yolov5s, for the task of vehicle damage detection. Various potential approaches, including transfer learning, image generation, and explainability analysis, were explored to achieve this goal. The transfer learning experiments revealed that none of the models achieved comparable mAP, precision, and recall performance to the baseline model, which was trained solely on the NRB dataset with the COCO pretrained model weights. The baseline model was also observed to outperform the model trained in a previous study. The difference could be attributed to a different data augmentation technique used, a different learning rate schedule, or a combination of these factors. Furthermore, the experiment results suggested that transfer learning is more effective when applied from a general task to a specific one. Then, to investigate the effectiveness of transfer learning further, it might be beneficial to transfer a pretrained model that hasn't reached optimality and implement a cross-validation procedure.

The use of synthetic image generation techniques to improve the data was also explored using the GAN and Stable Diffusion generative models. The results of the GAN experiments showed that while GANs are capable of generating images that resemble real vehicle images, the models in this study were not yet able to generate high-quality images that can be used for training a vehicle damage detector. However, future study might improve the performance of GANs by fine-tuning the training hyperparameters, testing other GAN architectures from the wide diversity of architectures that exists, and identifying more effective image quality evaluation metrics. Nevertheless, the results of the Stable Diffusion experiment were more promising. It was found that the model was able to generate high-quality images showing reconstructions of undamaged vehicles. However, the model sometimes lacked generative power to deal with damaged vehicles that had scattered silhouettes due to broken pieces that detached from the body of the vehicle. Moreover, it was found that the model was limited in its controllability and predictability, especially when given large generative power. Nevertheless, even heavily damaged vehicles could be converted into coherent reconstructed vehicles. The results of the model trained on the dataset augmented with the reconstructions were not *bad* enough to discard this investigation path, and further research is required to generate high-quality images. Thus, this path of investigation is still considered relevant. Moreover, it is noted that the model's performance on the scratches class penalised the model the most, which might be due to the quality of the synthetic reconstructions. The model might be further improved using different specific and constraining prompts. An additional explainability analysis on the synthetic data augmentated models to understand better how the model uses those augmentations.

Lastly, it was aimed to explain the model's decisions to gain trust and transparency in its decision process and find model limitations that might be addressed in future studies. The results showed that the model's attention was drawn to different regions of the image depending on the class of damage being detected. Moreover, it was found that the classification task is non-trivial due to the large inter-class similarity of damages, ambiguity in labelling, and large variety of damage shapes. It is thus suggested to work on mitigating those limitations. The confusion matrix revealed where the most confusion happened, and it was sometimes unclear how the model distinguished between similar severity classes, such as, scratches and side damages, and similar and neighbouring location classes. Some explanations might be found when exploring other models' layers to gain a deeper understanding of the model's decision-making process. It is also suggested to address the subjectivity and overlapping of damage labels by developing more discriminative damage classes. The model was also found to be struggling with excessive close-ups damage images and light and shadows reflections on the metallic vehicle bodies, which may hardly be solved except from discarding those images from the dataset, and which may be addressed by including a label for those lights and shadow reflections, respectively. More generally, creating a larger and more diverse dataset limiting the number of image contexts, and varying representations of the instances should be beneficial to the model performances.

In short, this study provides an investigation basis in transfer learning and synthetic data augmentation for future research and highlights the challenges of the damage detection task. Future studies could address the limitations derived from the explainability study to improve the model's performance in detecting damaged vehicles.

Bibliography

[1] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[2] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. *Face Aging With Conditional Generative Adversarial Networks*. 2017. DOI: 10.48550/ARXIV.1702.01983. URL: https://arxiv.org/abs/1702.01983.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. DOI: 10.48550/ARXIV.1701.07875. URL: https://arxiv.org/abs/1701.07875.

[4] Swapnali Bagal. *Detecting Vehicle Damage using Deep Learning*. https://algoanalytics.com/vehicleDamage.html.

[5] B. Bnspriyatham. *Car Damage Assessment using Deep Learning - Analytics Vidhya*. 2022. URL: https://medium.com/analytics-vidhya/car-damage-classification-using-deep-learning-d29fa1e9a520.

[6] Moritz Böhle et al. "Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer's Disease Classification". In: *Frontiers in Aging Neuroscience* 11 (July 2019). DOI: 10.3389/fnagi.2019.00194. URL: https://doi.org/10.3389%2Ffnagi.2019.00194.

[7] Ali Borji. *Pros and Cons of GAN Evaluation Measures*. 2018. DOI: 10.48550/ARXIV.1802.03446. URL: https://arxiv.org/abs/1802.03446.

[8] Jerome De Bruyn. "YOLOv5 , YOLOR Faster R-CNN : A Comparitive Study Between CNN Performances on Damaged Automobiles". In: (2022).

[9] Oriol Corcoll Andreu. *Semantic Image Cropping*. July 2021.

[10] Ekin D. Cubuk et al. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June.Section 3 (2019), pp. 113–123. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00020. arXiv: arXiv:1805.09501v3.

[11] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[12] Nicki Skafte Detlefsen et al. *TorchMetrics, Measuring Reproducibility in PyTorch*. Feb. 11, 2022. DOI: 10.21105/joss.04101. URL: %5Curl%7Bhttps://www.pytorchlightning.ai%7D.

[13] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. DOI: 10.48550/ARXIV.2105.05233. URL: https://arxiv.org/abs/2105.05233.

[14] Mahavir Dwivedi et al. *Deep Learning Based Car Damage Classification and Detection*. Sept. 2019. DOI: 10.13140/RG.2.2.18702.51525.

[15] Mark Everingham et al. "The Pascal Visual Object Classes (VOC) challenge". In: *International Journal of Computer Vision* 88 (June 2010), pp. 303–338. DOI: 10.1007/s11263-009-0275-4.

[16] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.4. Mar. 30, 2019. DOI: 10.5281/zenodo.3828935. URL: https://www.pytorchlightning.ai.

[17] Mohamed Mostafa Fouad et al. "Automated vehicle inspection model using a deep learning approach". In: *Journal of Ambient Intelligence and Humanized Computing* 0123456789 (2022). ISSN: 18685145. DOI: 10.1007/s12652-022-04105-3. URL: https://doi.org/10.1007/s12652-022-04105-3.

[18] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[19] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[20] Ben Graham et al. "LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference". In: *Proceedings of the IEEE International Conference on Computer Vision* (2021), pp. 12239–12249. ISSN: 15505499. DOI: 10.1109/ICCV48922.2021.01204. arXiv: 2104.01136.

[21] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. 2017. DOI: 10.48550/ARXIV.1704.00028. URL: https://arxiv.org/abs/1704.00028.

[22] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[23] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium". In: *CoRR* abs/1706.08500 (2017). arXiv: 1706.08500. URL: http://arxiv.org/abs/1706.08500.

[24] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2016. DOI: 10.48550/ARXIV.1611.07004. URL: https://arxiv.org/abs/1611.07004.

[25] Max Jaderberg et al. *Population Based Training of Neural Networks*. 2017. DOI: 10.48550/ARXIV.1711.09846. URL: https://arxiv.org/abs/1711.09846.

[26] Srimal Jayawardena. "Image Based Automatic Vehicle Damage Detection". In: November (2013).

[27] Glenn Jocher et al. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. Version v7.0. Nov. 2022. DOI: 10.5281/zenodo.7347926. URL: https://doi.org/10.5281/zenodo.7347926.

[28] Apostolos Karasmanoglou, Marios Antonakakis, and Michalis Zervakis. "Heatmap-based Explanation of YOLOv5 Object Detection with Layer-wise Relevance Propagation". In: June 2022, pp. 1–6. DOI: 10.1109/IST55454.2022.9827744.

[29] Jonathan Krause et al. "3D Object Representations for Fine-Grained Categorization". In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia, 2013.

[30] Sebastian Lapuschkin et al. “On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation”. In: *PLoS ONE* 10 (July 2015), e0130140. DOI: 10.1371/journal.pone.0130140.

[31] Richard Liaw et al. “Tune: A Research Platform for Distributed Model Selection and Training”. In: *arXiv preprint arXiv:1807.05118* (2018).

[32] Jae Hyun Lim and Jong Chul Ye. *Geometric GAN*. 2017. DOI: 10.48550/ARXIV.1705.02894. URL: https://arxiv.org/abs/1705.02894.

[33] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2014. DOI: 10.48550/ARXIV.1405.0312. URL: https://arxiv.org/abs/1405.0312.

[34] Scott Lundberg. *An introduction to explainable AI with Shapley values*. 2018. URL: https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%5C%20introduction%5C%20to%5C%20explainable%5C%20AI%5C%20with%5C%20Shaply%5C%20values.html (visited on 01/13/2022).

[35] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[36] Pooya Mohammadi Kazaj. *yolov5-gradcam*. github.com/pooya-mohammadi/yolov5-gradcam. 2021.

[37] Data Monsters. *A Quick Overview of Methods to Measure the Similarity Between Images*. https://medium.com/@datamonsters/a-quick-overview-of-methods-to-measure-the-similarity-between-images-f907166694ee. 2020W.

[38] Neokt. *Damage Image Dataset*. https://github.com/neokt/car-damage-detective. Open Source Dataset. visited on 2022-12-25. Dec. 2017. URL: https://github.com/neokt/car-damage-detective.

[39] Kalpesh Patil, Mandar Kulkarni, and Shirish Karande. “DEEP LEARNING BASED CAR DAMAGE CLASSIFICATION Kalpesh Patil Mandar Kulkarni Shirish Karande TCS Innovation Labs , Pune , India”. In: (2018), pp. 3–7. URL: https://www.ee.iitb.ac.in/student/$%5Csim$kalpesh.patil/material/car_damage.pdf.

[40] Kalpesh Patil, Mandar Kulkarni, Anand Sriraman, et al. “Deep learning based car damage classification”. In: *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017* 2017-December.September (2017), pp. 50–54. DOI: 10.1109/ICMLA.2017.0-179.

[41] Gabrielle Ras et al. *Explainable Deep Learning: A Field Guide for the Uninitiated*. 2020. DOI: 10.48550/ARXIV.2004.14545. URL: https://arxiv.org/abs/2004.14545.

[42] Reply. *Assessing car damage with the use of image recognition*. https://www.reply.com/en/data-world/assessing-car-damage-with-the-use-of-image-recognition. May 2013.

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-Resolution Image Synthesis With Latent Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695.

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. "High-Resolution Image Synthesis with Latent Diffusion Models". In: (2022), pp. 10674–10685. ISSN: 10636919. DOI: `10.1109/cvpr52688.2022.01042`. arXiv: `2112.10752`.

[45] R.E. van Ruitenbeek and S. Bhulai. "Convolutional Neural Networks for vehicle damage detection". In: *Machine Learning with Applications* 9.May (2022), p. 100332. ISSN: 26668270. DOI: `10.1016/j.mlwa.2022.100332`. URL: `https://doi.org/10.1016/j.mlwa.2022.100332`.

[46] Chitwan Saharia, William Chan, Huiwen Chang, et al. *Palette: Image-to-Image Diffusion Models.* 2021. DOI: `10.48550/ARXIV.2111.05826`. URL: `https://arxiv.org/abs/2111.05826`.

[47] Chitwan Saharia, William Chan, Saurabh Saxena, et al. *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding.* 2022. arXiv: `2205.11487 [cs.CV]`.

[48] Chitwan Saharia, Jonathan Ho, et al. *Image Super-Resolution via Iterative Refinement.* 2021. DOI: `10.48550/ARXIV.2104.07636`. URL: `https://arxiv.org/abs/2104.07636`.

[49] Sammy. *Damaged Vehicle Images Dataset.* `https://universe.roboflow.com/sammy/damaged-vehicle-images`. Open Source Dataset. visited on 2022-12-25. June 2022. URL: `https://universe.roboflow.com/sammy/damaged-vehicle-images`.

[50] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. DOI: `10.1007/s11263-019-01228-7`. URL: `https://doi.org/10.1007%2Fs11263-019-01228-7`.

[51] Sriram Sridhar78. *sriram-car-classifier.* `https://huggingface.co/SriramSridhar78/sriram-car-classifier?doi=true`. 2021.

[52] Xuan Su et al. *Dual Diffusion Implicit Bridges for Image-to-Image Translation.* 2022. DOI: `10.48550/ARXIV.2203.08382`. URL: `https://arxiv.org/abs/2203.08382`.

[53] Yuki Tatsunami and Masato Taki. "Sequencer: Deep LSTM for Image Classification". In: NeurIPS (2022), pp. 1–26. arXiv: `2205.01972`. URL: `http://arxiv.org/abs/2205.01972`.

[54] Hugo Touvron, Matthieu Cord, Matthijs Douze, et al. "Training data-efficient image transformers distillation through attention". In: (2020), pp. 1–22. arXiv: `2012.12877`. URL: `http://arxiv.org/abs/2012.12877`.

[55] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, et al. "Going deeper with Image Transformers". In: *Proceedings of the IEEE International Conference on Computer Vision* (2021), pp. 32–42. ISSN: 15505499. DOI: `10.1109/ICCV48922.2021.00010`. arXiv: `2103.17239`.

[56] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: `10.1109/TIP.2003.819861`.

[57] Bichen Wu et al. *Visual Transformers: Token-based Image Representation and Processing for Computer Vision.* 2020. arXiv: `2006.03677 [cs.CV]`.

[58] Jaechang Yoo, Heesong Eom, and Yong Choi. "Image-To-Image Translation Using a Cross-Domain Auto-Encoder and Decoder". In: *Applied Sciences* 9 (Nov. 2019), p. 4780. DOI: `10.3390/app9224780`.

[59] Ning Yu et al. *Dual Contrastive Loss and Attention for GANs*. 2021. DOI: `10.48550/ARXIV.2103.16748`. URL: `https://arxiv.org/abs/2103.16748`.

[60] Haidi Zhu et al. "A Review of Video Object Detection: Datasets, Metrics and Methods of video object detection; deep learning-based video object detection". In: *Applied Sciences* 10 (Nov. 2020), p. 7834. DOI: `10.3390/app10217834`.

[61] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2017. DOI: `10.48550/ARXIV.1703.10593`. URL: `https://arxiv.org/abs/1703.10593`.

[62] Chubin Zhuang, Zhen Lei, and Stan Z. Li. *SADet: Learning An Efficient and Accurate Pedestrian Detector*. 2020. DOI: `10.48550/ARXIV.2007.13119`. URL: `https://arxiv.org/abs/2007.13119`.

Appendix

A.1 NEOKT Dataset Labelling

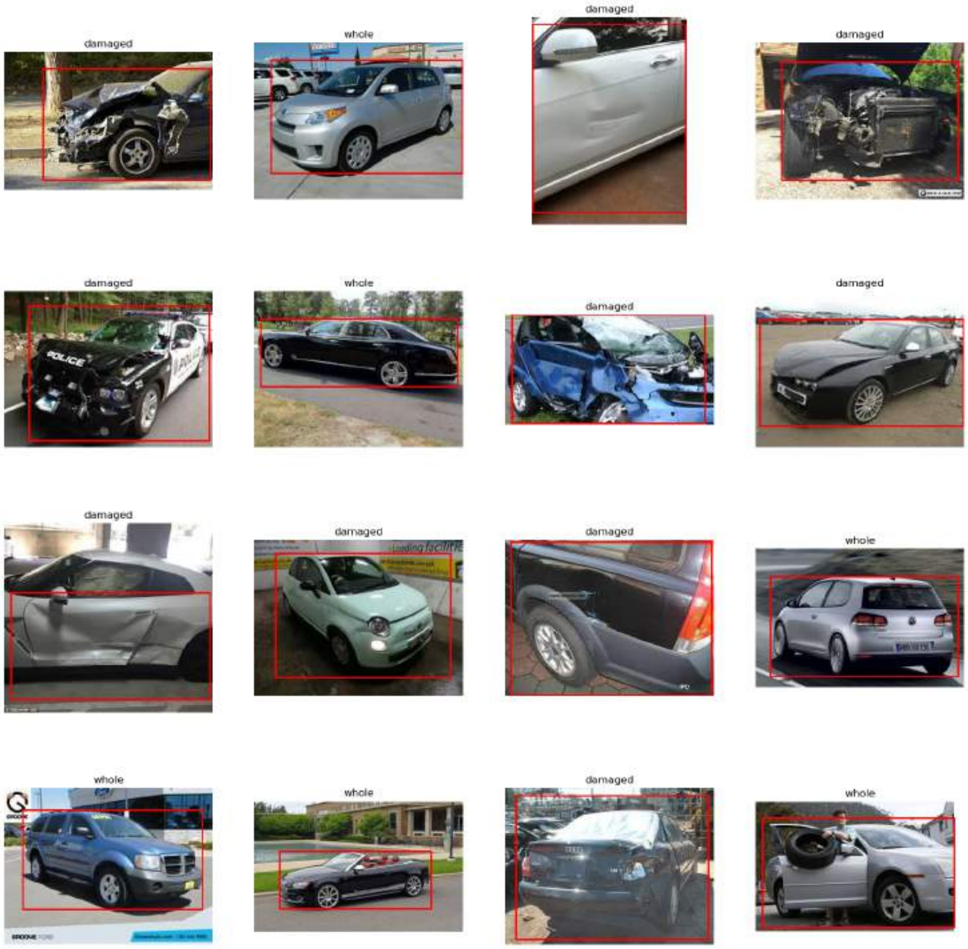
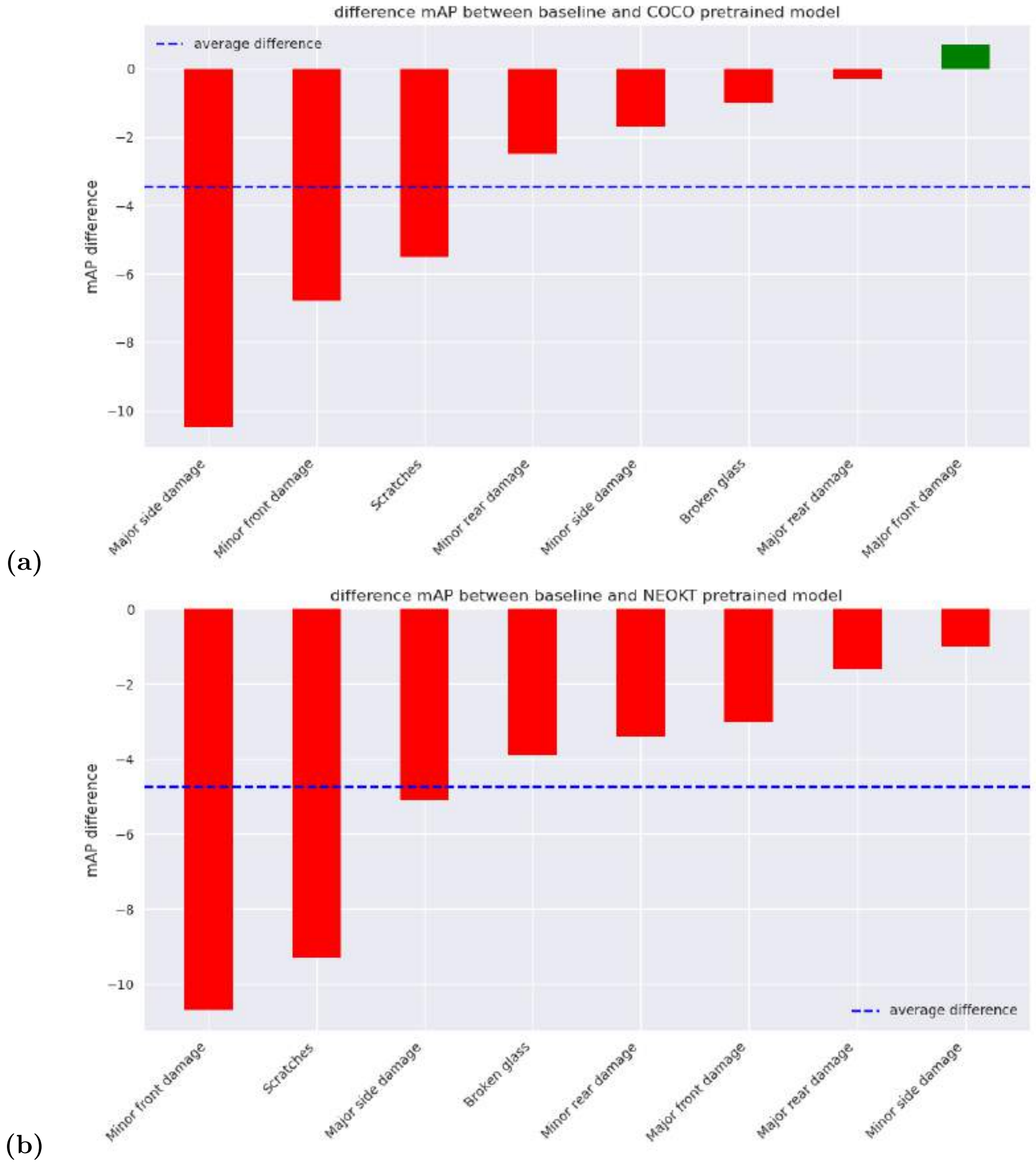


Figure A.1: Since NEOKT was a classification dataset of vehicles with damaged-whole classes, bounding box information was missing to make the dataset a detection dataset. This information was added using a Yolov5 model, detecting car objects on the images. Yolo's weights are set as the pretrained COCO weights, which is a benchmark detection data containing about 100.000 car objects Corcoll Andreu 2021. Yolo can confidently detect bounding boxes of whole cars however it can be observed in this figure that it is not always the case for damaged cars, especially for close-ups of the vehicles.

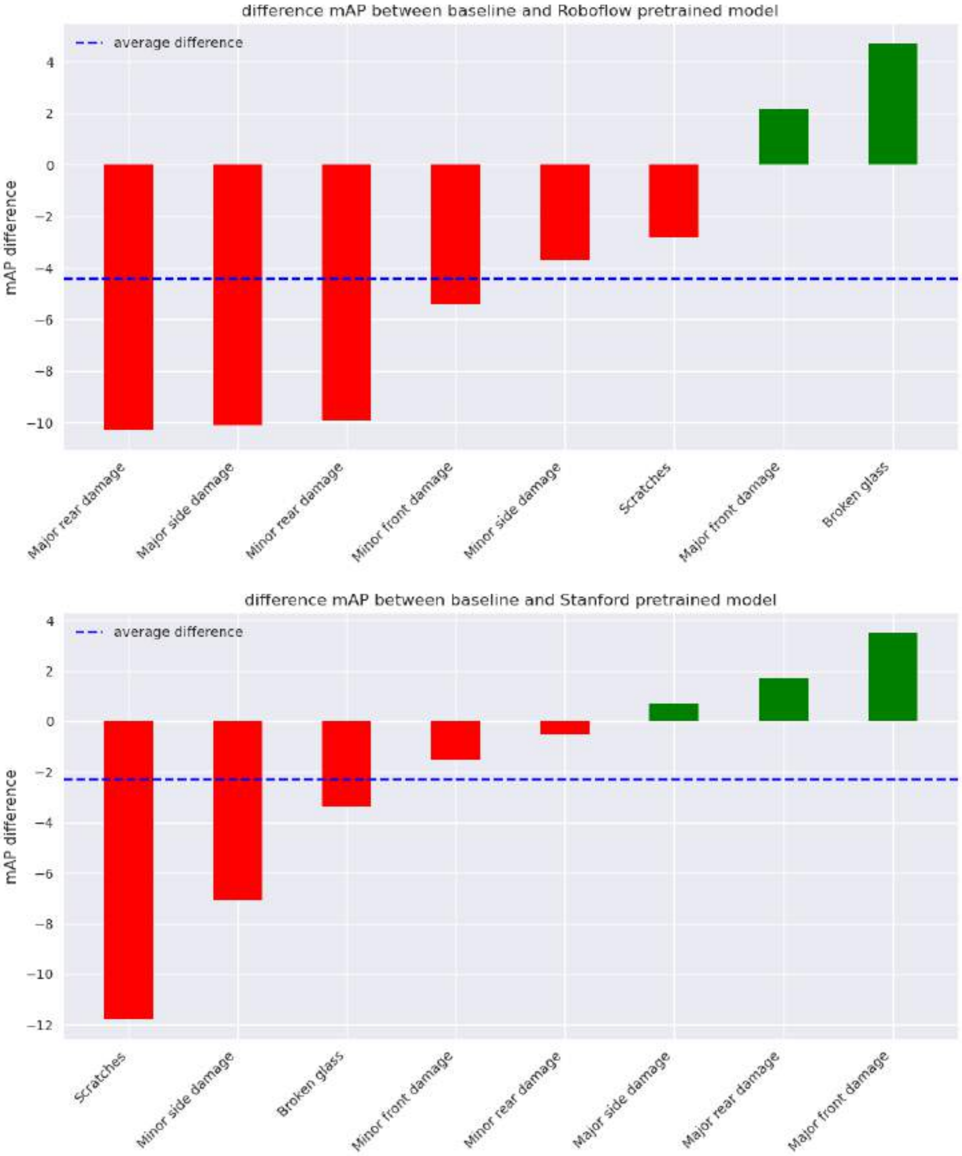
A.2 Baseline and Fine-tuned models mAP per class Comparisons



(a)

(b)

Figure A.2: Damage class-specific differences in test mAPs between the baseline model and the coco pretrained model (a) and the NEOKT pretrained model (b)



(c)

(d)

Figure A.3: Continued damage class-specific differences in test mAPs between the baseline model and the roboflow pretrained model (c) and the stanford cars pretrained model (d)

A.3 NRB dataset test dataset samples of damaged cars, and their generated whole reconstructions, generated with a Diffusion Model

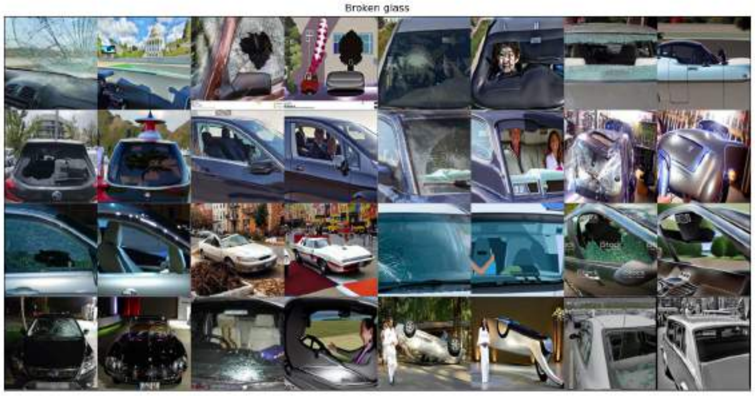


Figure A.4: test dataset samples of the Broken Glass class

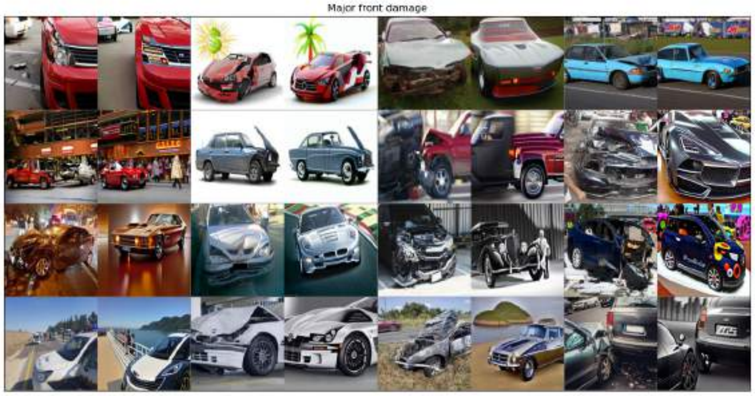


Figure A.5: test dataset samples of the Major front damage class

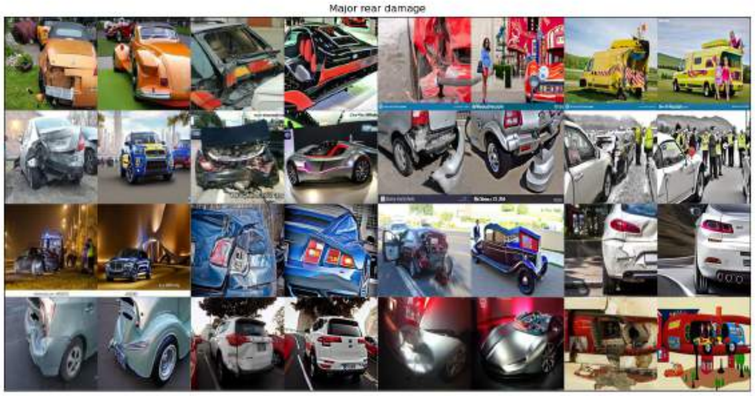


Figure A.6: test dataset samples of the Major rear damage class

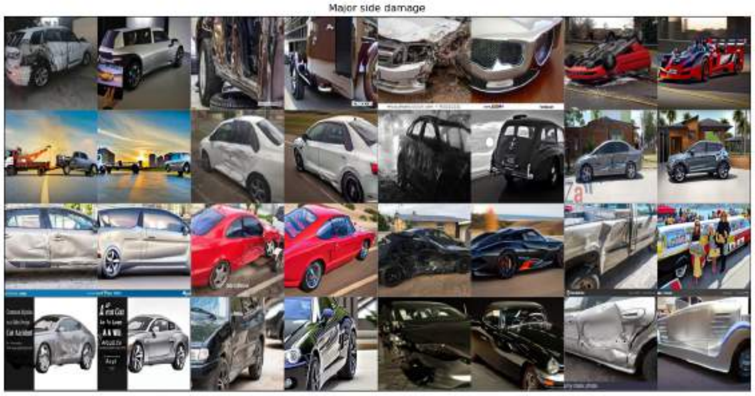


Figure A.7: test dataset samples of the Major side damage class

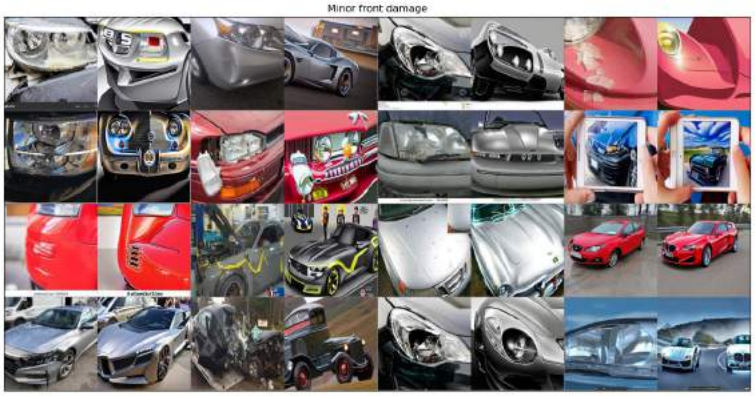


Figure A.8: test dataset samples of the Minor front damage class

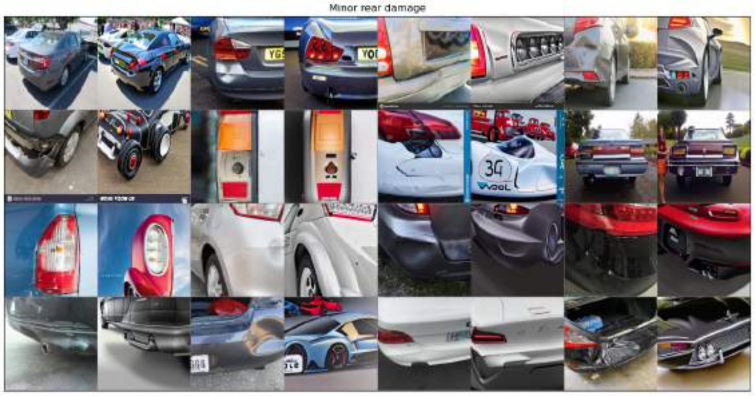


Figure A.9: test dataset samples of the Minor rear damage class

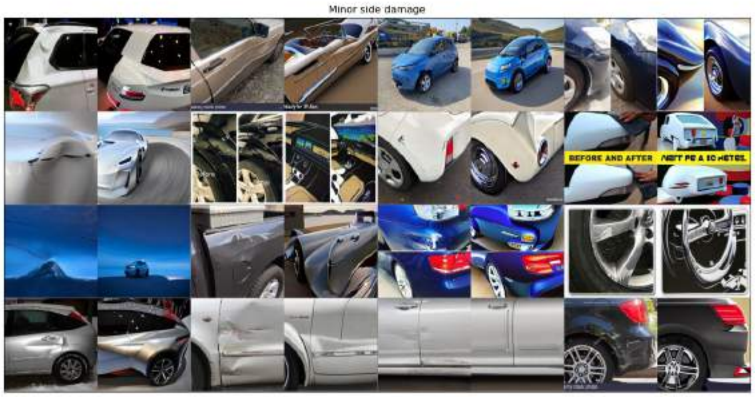


Figure A.10: test dataset samples of the Minor side damage class

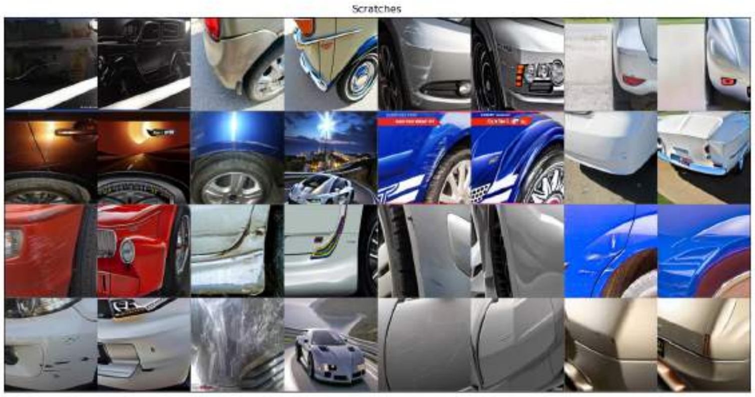
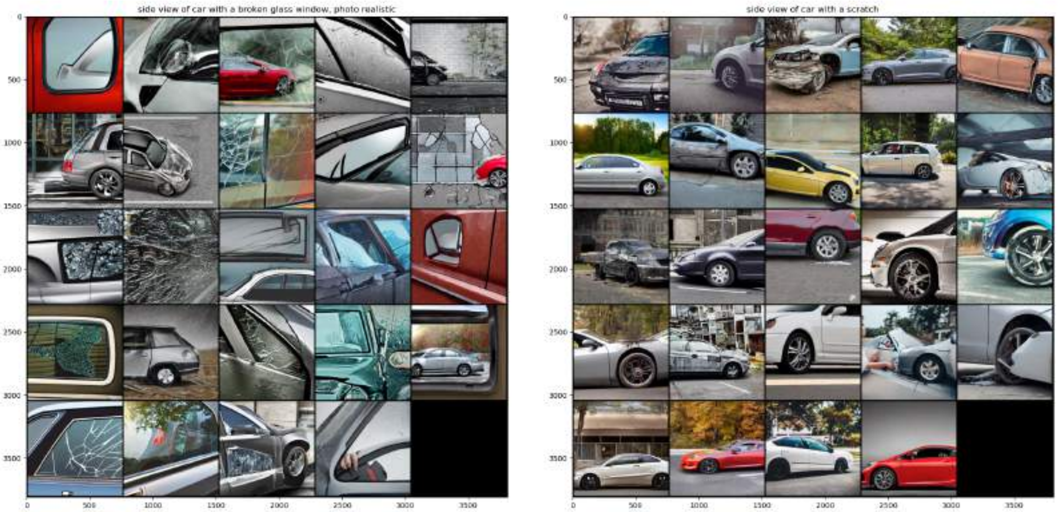


Figure A.11: test dataset samples of the Scratches class

A.4 Synthetic Dataset


side view of car with a broken glass window, photo realistic
side view of car with a scratch
0
500
1000
1500
2000
2500
3000
3500

(a)

(b)

Figure A.12: Two synthetically generated batches of the prompt (a) "side view of a car with a broken glass window, photo-realistic" and (b) "side view of a car with a scratch", using a Diffusion Model

A.5 Exai Heatmap Visualisations

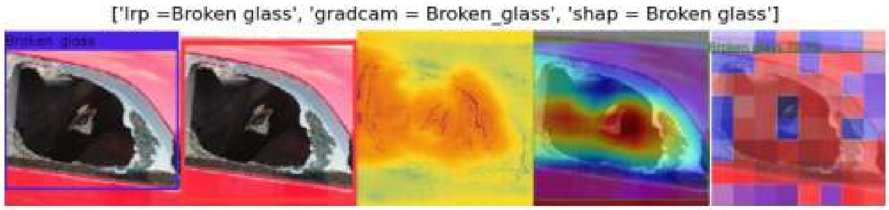

['lrp =Broken glass', 'gradcam = Broken_glass', 'shap = Broken glass']
Broken glass

Figure A.13: broken glass damage

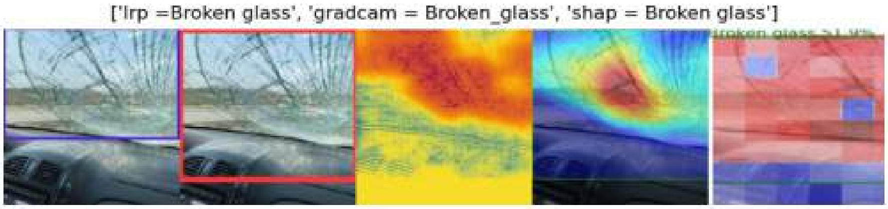

['lrp =Broken glass', 'gradcam = Broken_glass', 'shap = Broken glass']

Figure A.14: broken glass damage

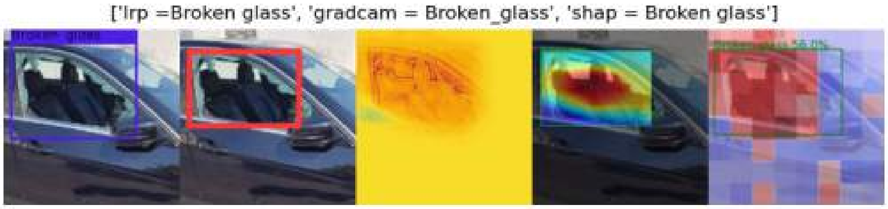

['lrp =Broken glass', 'gradcam = Broken_glass', 'shap = Broken glass']
Broken glass

Figure A.15: broken glass damage

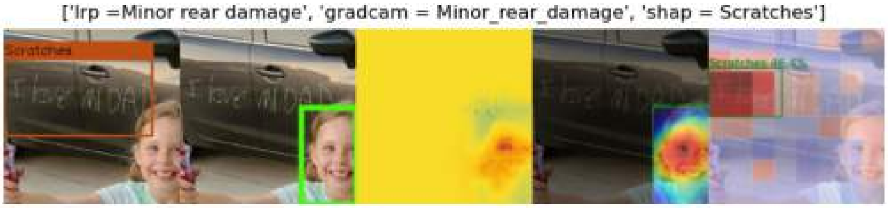

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Scratches']
Scratches

Figure A.16: wrong detection

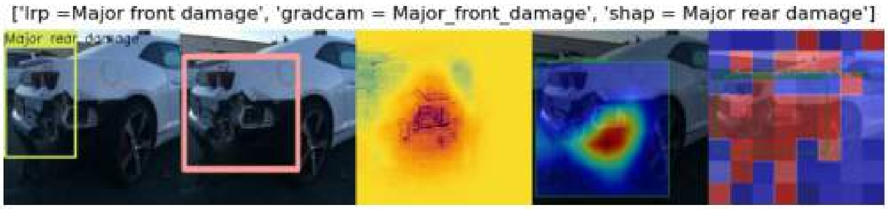

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major rear damage']
Major rear damage

Figure A.17: wrong detection, front instead of rear damage

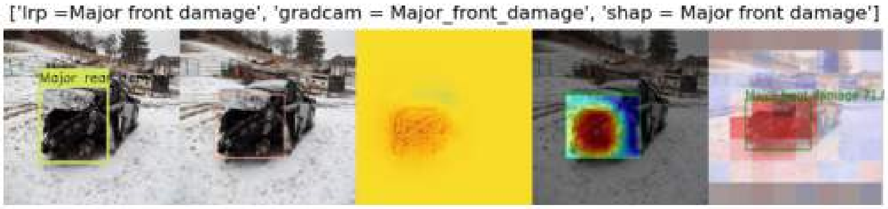

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major front damage']
Major_rear

Figure A.18: wrong detection, front instead of rear damage

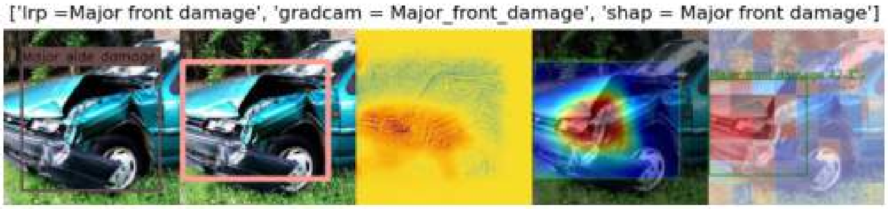

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major front damage']
Major_side_damage

Figure A.19: wrong detection, front instead of side damage

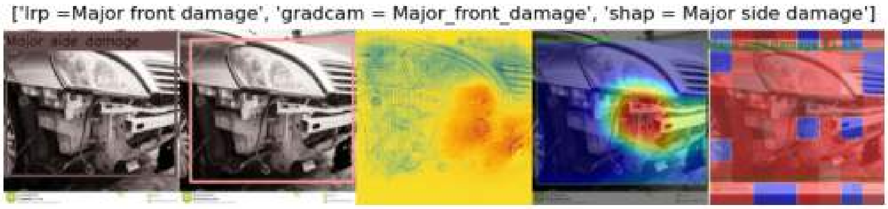

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major side damage']
Major_side_damage

Figure A.20: wrong detection, front instead of side damage

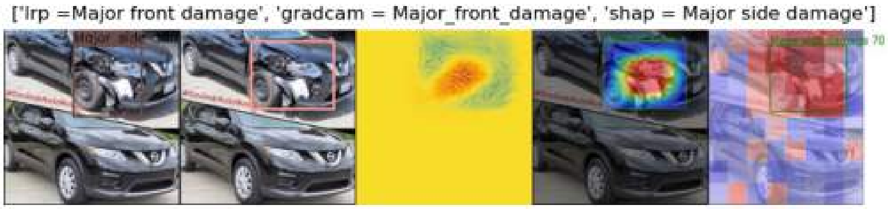

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major side damage']
Major_side

Figure A.21: wrong detection, front instead of side damage

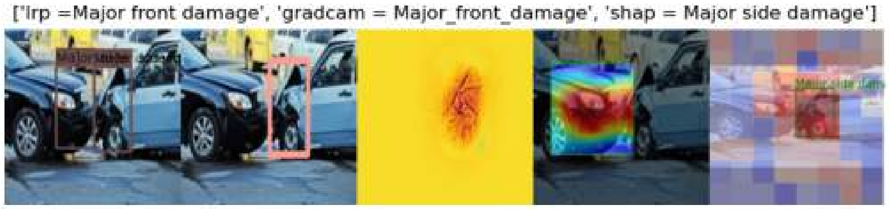

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major side damage']
Major side dam

Figure A.22: wrong detection, front instead of side damage

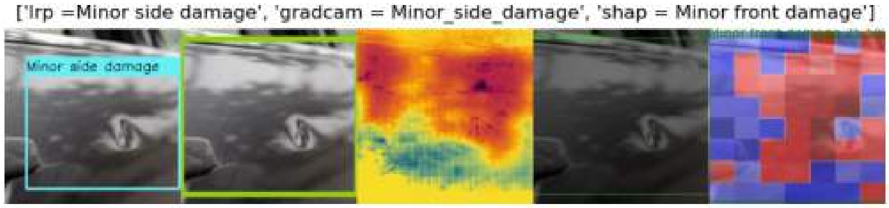

['lrp =Minor side damage', 'gradcam = Minor_side_damage', 'shap = Minor front damage']
Minor side damage

Figure A.23: wrong detection, the damage is in reality a light reflection

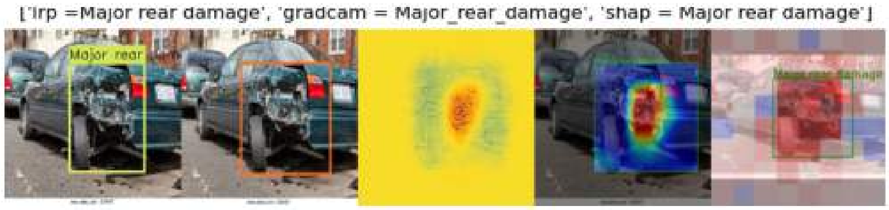

['lrp =Major rear damage', 'gradcam = Major_rear_damage', 'shap = Major rear damage']
Major rear
Major rear damage

Figure A.24: major front damage

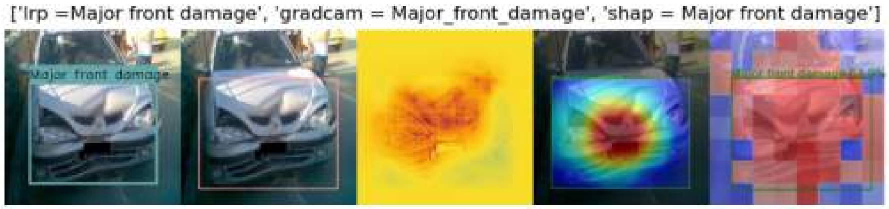

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major front damage']
Major front damage

Figure A.25: major front damage

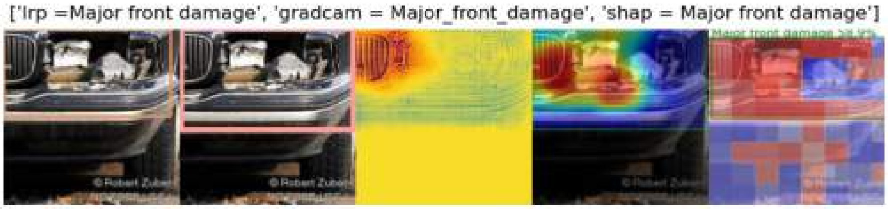

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major front damage']
©Robert Zuber

Figure A.26: major front damage

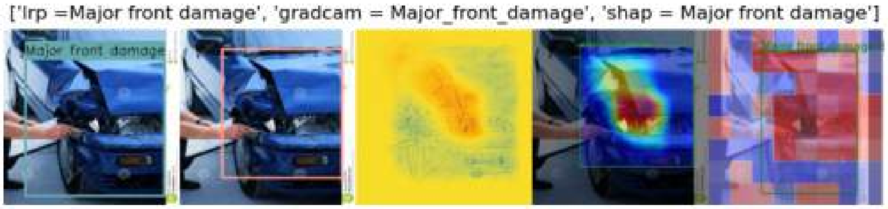

['lrp =Major front damage', 'gradcam = Major_front_damage', 'shap = Major front damage']
Major_front_damage

Figure A.27: major front damage

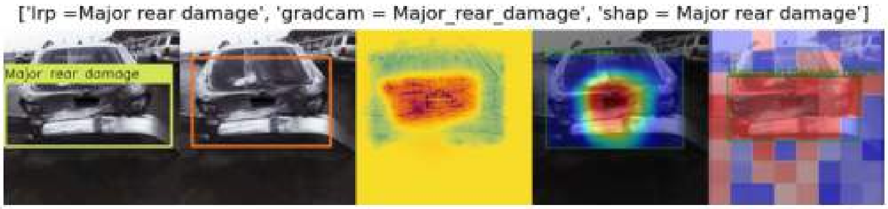

['lrp =Major rear damage', 'gradcam = Major_rear_damage', 'shap = Major rear damage']
Major_rear_damage

Figure A.28: major rear damage

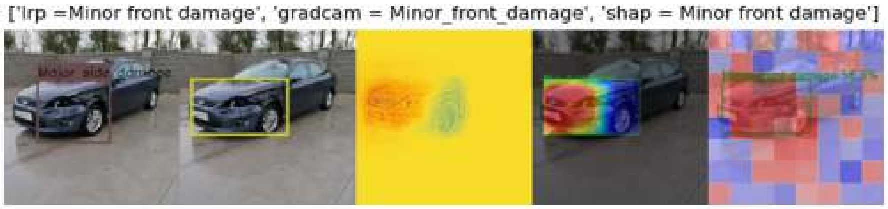

['lrp =Minor front damage', 'gradcam = Minor_front_damage', 'shap = Minor front damage']
Major_side_damage

Figure A.29: wrong detection, minor front instead of major side damage

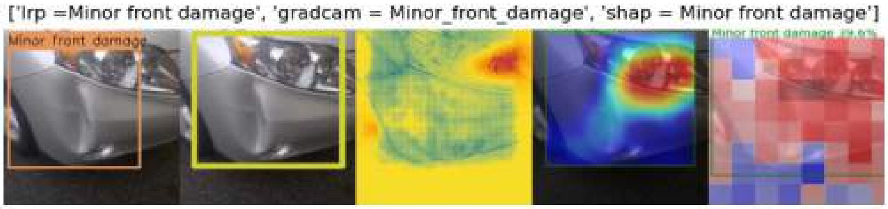

['lrp =Minor front damage', 'gradcam = Minor_front_damage', 'shap = Minor front damage']
Minor front damage
Minor front damage

Figure A.30: minor front damage

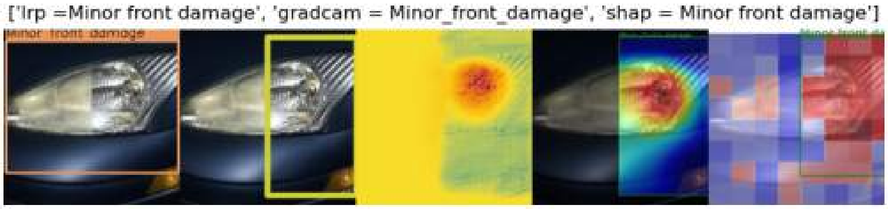

['lrp =Minor front damage', 'gradcam = Minor_front_damage', 'shap = Minor front damage']
Minor front damage

Figure A.31: minor front damage

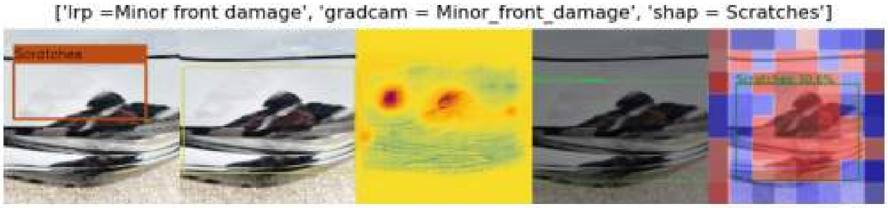

['lrp =Minor front damage', 'gradcam = Minor_front_damage', 'shap = Scratches']
Scratches

Figure A.32: wrong detection, minor front instead of scratches damage

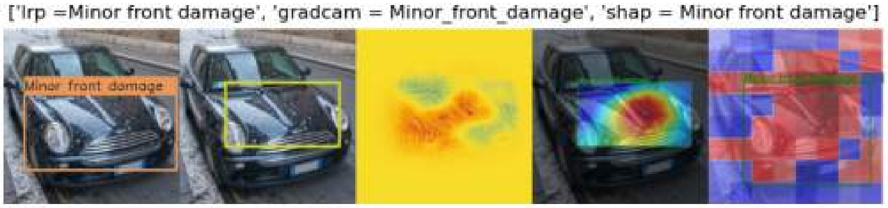

['lrp =Minor front damage', 'gradcam = Minor_front_damage', 'shap = Minor front damage']
Minor front damage

Figure A.33: minor front minor damage

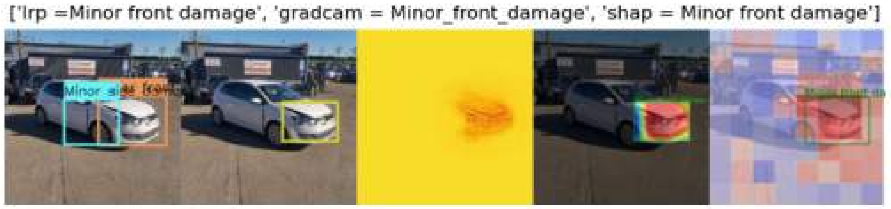

['lrp =Minor front damage', 'gradcam = Minor_front_damage', 'shap = Minor front damage']
Minor side

Figure A.34: minor front damage

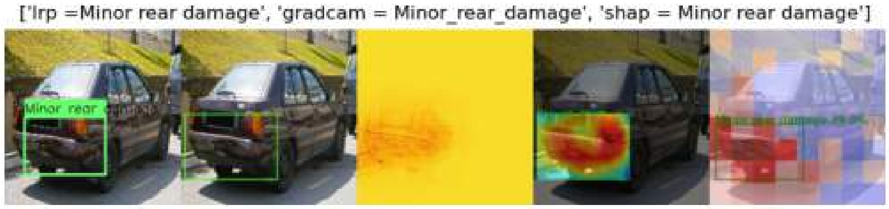

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']
Minor rear

Figure A.35: minor rear damage

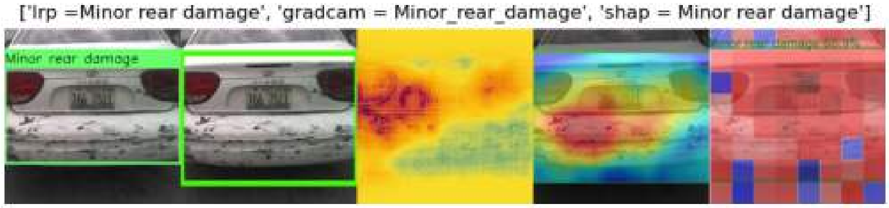

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']
Minor rear damage

Figure A.36: minor rear damage

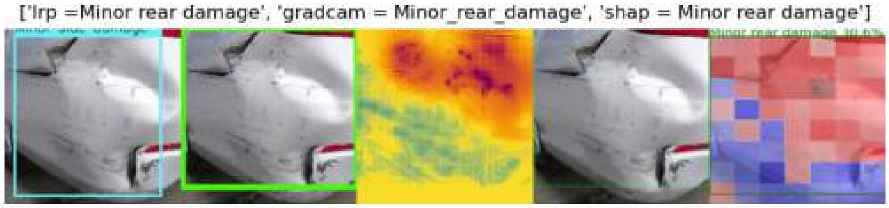

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']
Minor side damage
Minor rear damage

Figure A.37: minor rear damage

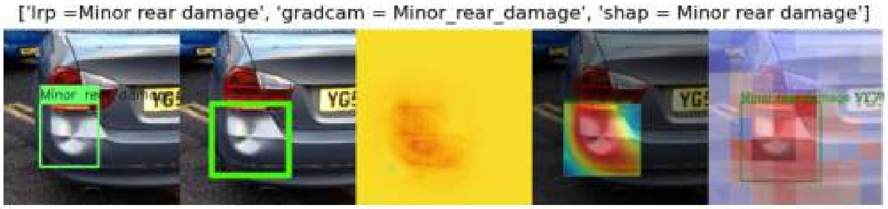

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']
Minor rear damage
YG5
Minor rear damage

Figure A.38: minor rear damage

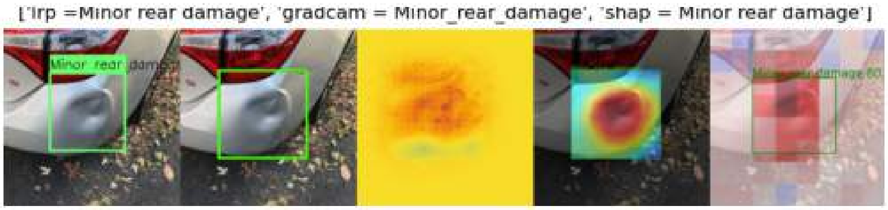

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']
Minor rear damage
Minor rear damage 80

Figure A.39: minor rear damage

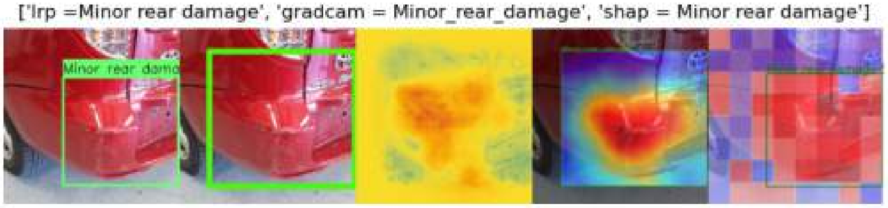

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']
Minor rear dama

Figure A.40: minor rear damage

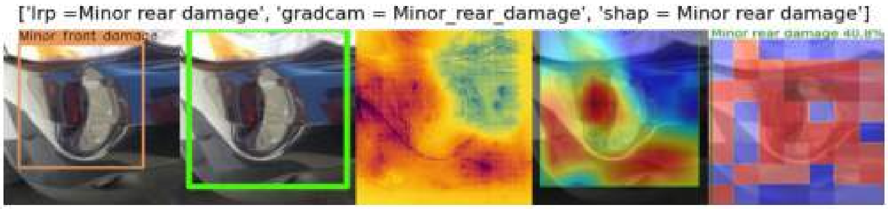

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']
Minor front damage
Minor rear damage 40.8%

Figure A.41: wrong detection, minor rear instead of minor front damage

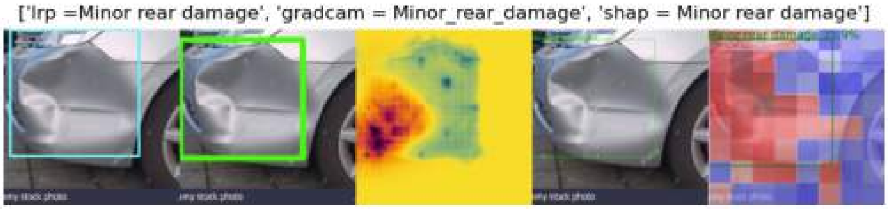

['lrp =Minor rear damage', 'gradcam = Minor_rear_damage', 'shap = Minor rear damage']

Figure A.42: wrong detection, minor rear instead of minor side damage

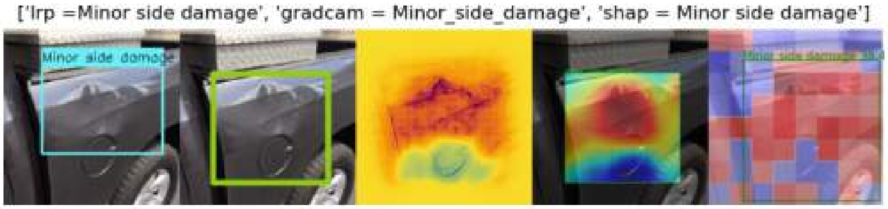

['lrp =Minor side damage', 'gradcam = Minor_side_damage', 'shap = Minor side damage']
Minor side damage

Figure A.43: minor side damage

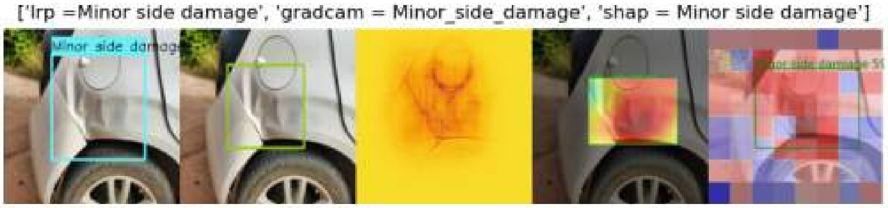

['lrp =Minor side damage', 'gradcam = Minor_side_damage', 'shap = Minor side damage']
Minor side damage

Figure A.44: minor side damage

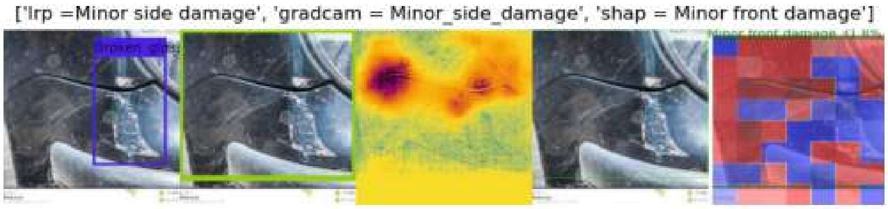

['lrp =Minor side damage', 'gradcam = Minor_side_damage', 'shap = Minor front damage']
broken glass

Figure A.45: wrong detection, minor side instead of broken glass damage

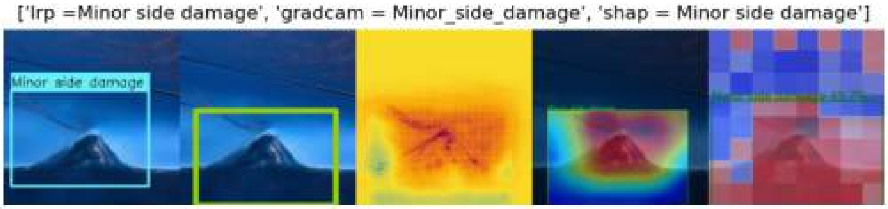

['lrp =Minor side damage', 'gradcam = Minor_side_damage', 'shap = Minor side damage']
Minor side damage

Figure A.46: minor side damage

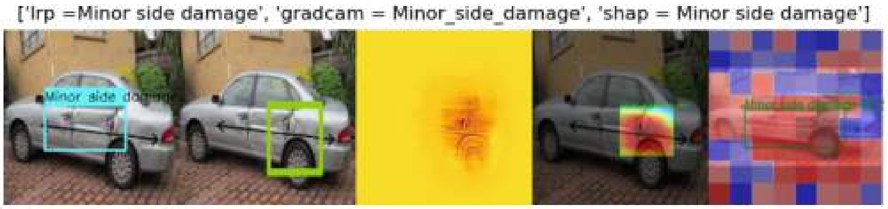

['lrp =Minor side damage', 'gradcam = Minor_side_damage', 'shap = Minor side damage']
Minor side damage

Figure A.47: minor side damage

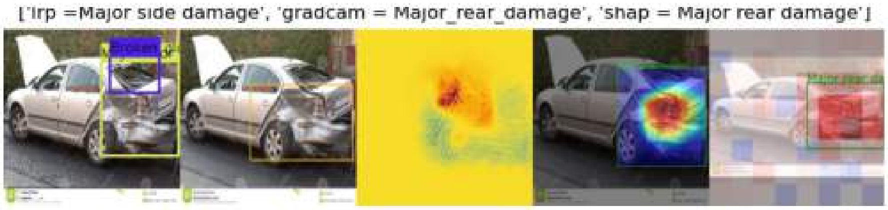

['lrp =Major side damage', 'gradcam = Major_rear_damage', 'shap = Major rear damage']
Broken

Figure A.48: wrong detection, rear instead of side damage

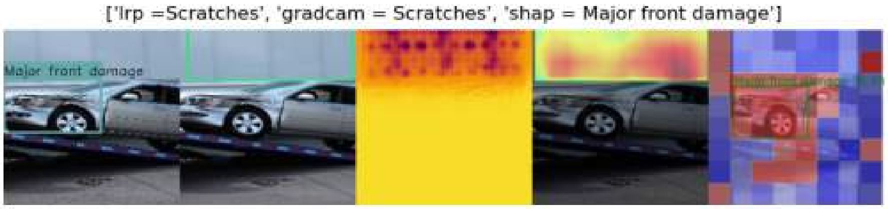

['lrp =Scratches', 'gradcam = Scratches', 'shap = Major front damage']
Major front damage

Figure A.49: wrong detection, artefact in the background detected as scratches damage

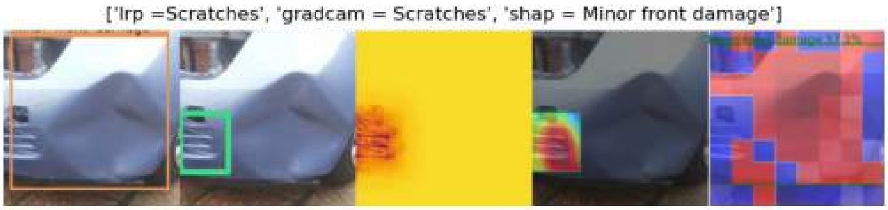

['lrp =Scratches', 'gradcam = Scratches', 'shap = Minor front damage']

Figure A.50: wrong detection, car radiator detected as scratches damage

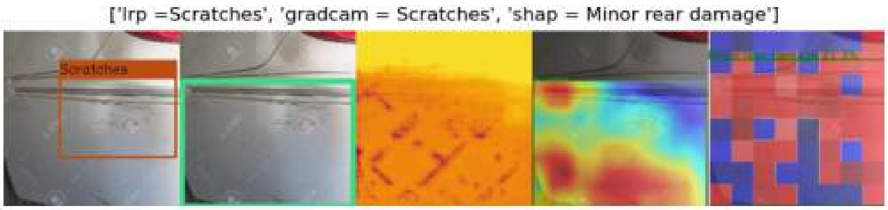

['lrp =Scratches', 'gradcam = Scratches', 'shap = Minor rear damage']
Scratches

Figure A.51: wrong detection, scratches damage instead of some image artifact

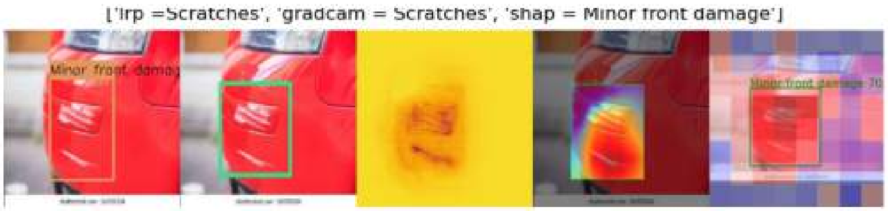

['lrp =Scratches', 'gradcam = Scratches', 'shap = Minor front damage']
Minor front damag
Minor front damage 70

Figure A.52: wrong detection scratches, instead of minor front damage

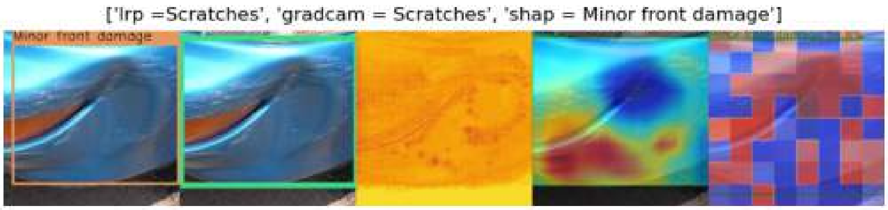

['lrp =Scratches', 'gradcam = Scratches', 'shap = Minor front damage']
Minor front damage

Figure A.53: wrong detection scratches, instead of minor front damage

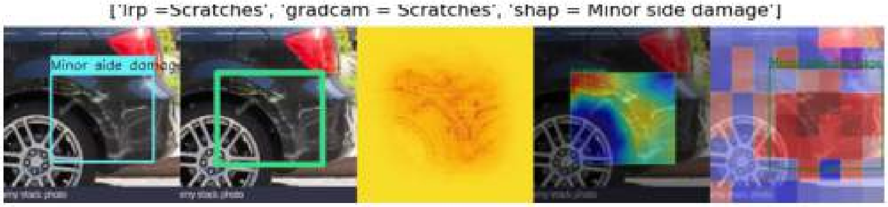

['lrp =Scratches', 'gradcam = Scratches', 'shap = Minor side damage']
Minor side damag

Figure A.54: wrong detection scratches, instead of minor side damage

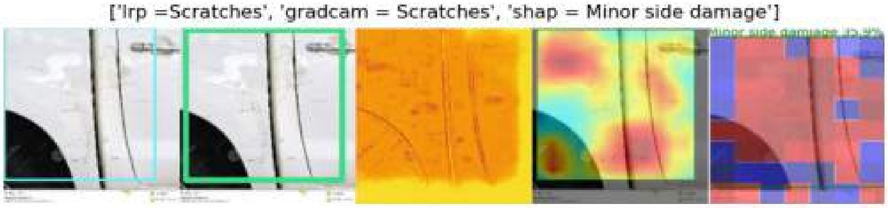

['lrp =Scratches', 'gradcam = Scratches', 'shap = Minor side damage']
Minor side damage

Figure A.55: wrong detection, scratches instead of minor side damage

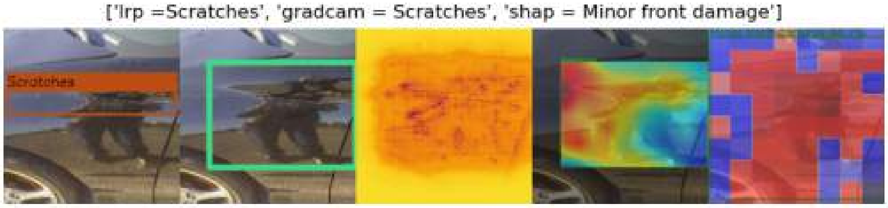

['lrp =Scratches', 'gradcam = Scratches', 'shap = Minor front damage']
Scratches

Figure A.56: scratches damage

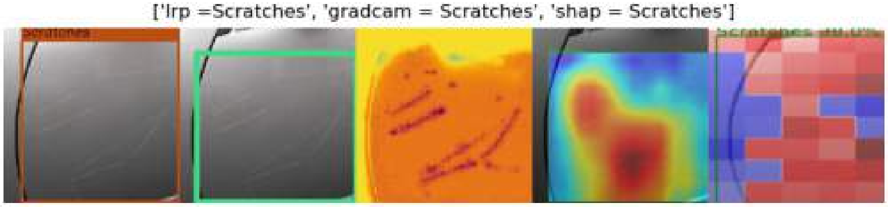

['lrp =Scratches', 'gradcam = Scratches', 'shap = Scratches']
Scratches
Scratches

Figure A.57: scratches damage

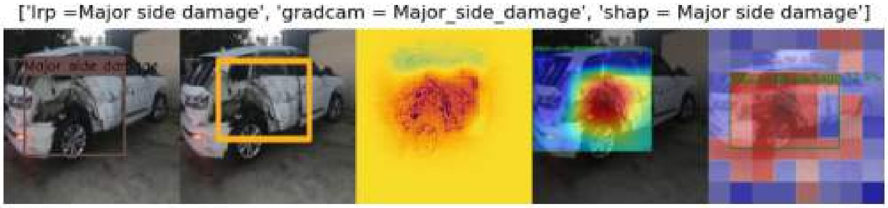

['lrp =Major side damage', 'gradcam = Major_side_damage', 'shap = Major side damage']

Figure A.58: side damage

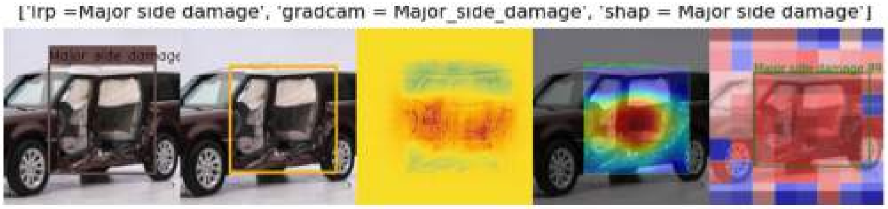

['lrp =Major side damage', 'gradcam = Major_side_damage', 'shap = Major side damage']
Major_side_damage

Figure A.59: side damage

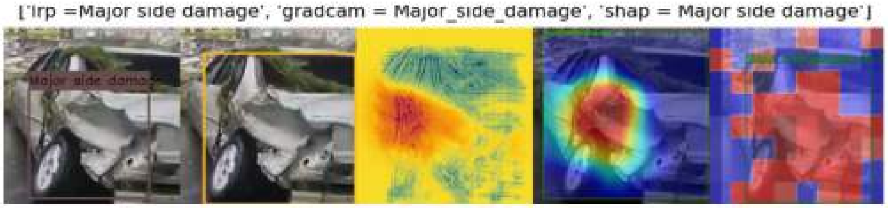

['lrp =Major side damage', 'gradcam = Major_side_damage', 'shap = Major side damage']
Major_side_damage

Figure A.60: side damage

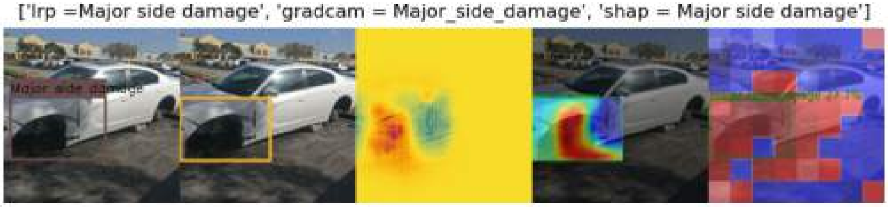

['lrp =Major side damage', 'gradcam = Major_side_damage', 'shap = Major side damage']
Major_side_damage

Figure A.61: side damage