electronics

MDPI

Article

An Insurtech Platform to Support Claim Management Through the Automatic Detection and Estimation of Car Damage from Pictures

Mohab Mahdy Helmy Atanasious [1], Valentina Becchetti [1], Alessandro Giuseppi [1], Antonio Pietrabissa [1], Vito Arconzo [2], Gerardo Gorga [2], Gonzalo Gutierrez [2], Ahmed Omar [2], Marco Pietrini [2], Meher Anvesh Rangisetty [2], Lorenzo Ricciardi Celsi [2,*], Federico Santini [2] and Enrico Scianaro [2]

[1] Dipartimento di Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti, Sapienza Universita di Roma, Via Ariosto 25, 00185 Rome, Italy; atanasious@diag.uniroma1.it (M.M.H.A.); v.becchetti@diag.uniroma1.it (V.B.); giuseppi@diag.uniroma1.it (A.G.); pietrabissa@diag.uniroma1.it (A.P.)

[2] Whoosnap Srl, Via Marsala 29H/I, 00185 Rome, Italy; gonzalo.gutierrez@insoore.com (G.G.); meher.rangisetty@insoore.com (M.A.R.)

* Correspondence: lorenzo.ricciardicelsi@insoore.com

Abstract: Claims management is a complex process through which an insurance company or responsible entity addresses and handles compensation requests from policyholders who have suffered damage or losses. This process entails several stages, including the notification of the claim, damage assessment, settlement of compensation, and, if necessary, dispute resolution. Fair, transparent and timely claims management is crucial for maintaining policyholders' trust while also limiting the financial impact on the insurer. Technological innovations, such as the use of artificial intelligence and automation, are positively influencing this sector, enabling faster and more effective claims management. This study reports on Insoore AI, an insurtech solution that aims to automate a portion of claims management by integrating a computer vision solution based on some latest developments in deep learning to automatically recognize and localize car damage from user-provided pictures.

Keywords: insurance claims management; deep learning; image segmentation; damage recognition; damage detection; computer vision

check for updates

Citation: Atanasious, M.M.H.; Becchetti, V.; Giuseppi, A.; Pietrabissa, A.; Arconzo, V.; Gorga, G.; Gutierrez, G.; Omar, A.; Pietrini, M.; Rangisetty, M.A.; et al. An Insurtech Platform to Support Claim Management Through the Automatic Detection and Estimation of Car Damage from Pictures. *Electronics* **2024**, *13*, 4333. https://doi.org/10.3390/electronics13224333

Academic Editor: Jian Sun

Received: 26 September 2024
Revised: 22 October 2024
Accepted: 1 November 2024
Published: 5 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Overview

The insurance industry is being revolutionized by the new players and products that leverage the latest developments of artificial intelligence (AI) and their unprecedented capabilities in analysing and processing large quantities of data [1]. Due to the criticality and impact on society of insurance, integrating advanced technological solutions into the insurance market (insurtech) to develop new solutions has to cope with existing regulations and legal frameworks. In fact, each national market may require clients' data to be treated differently and may prevent the automation of certain aspects or phases of the insurance claim process. The present paper focuses on an insurtech solution, dubbed Insoore AI, developed for the Italian market that integrates advanced computer vision technologies to automate a significant portion of the damage identification and repair cost estimation. The proposed system represents a pipeline for the process of estimating the repair and the action to perform for damaged pieces of a car after an accident. The pipeline makes use only of the images sent by the client.

The remainder of the paper is organized as follows: in Section 2, a background of the Italian insurance market is presented. In contrast, the technical background related to computer vision and the use of AI in vehicle damage detection is illustrated in Section 3. The proposed AI pipeline of this work is described in Section 4. Finally, the results of the proposed pipeline and the conclusions are reported in Sections 5 and 6 respectively.

2. Background of the Italian Insurance Market and Innovation of the Proposed Solution

Car insurance constitutes one of the largest markets globally. When a motorist is involved in an accident, the compensation claim procedure is often protracted and complex. The insurance company's damage coverage process commences by filling the documentation at the accident site and then involves the capture of photographs of the vehicle's damaged parts to facilitate the damage analysis by the insurance companies.

In Italy, several specific laws regulate the insurance system for road accidents to ensure adequate coverage for damage resulting from collisions or accidents. Motor Third Party Liability Insurance (RCA, "Responsabilità Civile Autoveicoli") is the mandatory coverage set by law that covers damage caused to third parties in the event of a road accident. RCA compensates for material damage and injuries caused to other vehicles or individuals involved in the accident, and all drivers must have it.

In the event of an accident, there are several steps to carry out before taking any insurance action:

- The involved drivers must immediately notify their insurance company of the incident within 3 days (this report can be made by phone, internet, or through the traditional registered letter with return receipt using specific forms).
- Detailed information about the incident, including vehicle details, road conditions, the position of the vehicles involved in the accident (photos could be very useful), when they are available for inspection by the insurance experts, and witness names, must be provided to the insurance company.
- It is necessary to specify the date, time, and location of the incident, the licence plates of the involved vehicles, the policy under which one is protected, the counterpart's details, and the insurance company used.
- To streamline the whole process, it is useful to fill out a standardized document called "Constatazione Amichevole di Incidente" (Friendly Accident Statement) or "Convenzione Indennizzo Diretto CID" with the counterpart's signature or a single signature, in case an agreement on the accident description is not reached, containing all the necessary information for reporting the claim.
- In the event of injuries, the injured parties' details and the extent of the injury must be specified. Medical reports immediately following the incident must also be included in the report.

Only once all the information is collected will the insurance company start to analyse it to determine liability and assess the damage suffered.

The insurance expert is the figure responsible for this procedure; its role was established by the Italian Law of 17 February 1992 n. 166 [2]. The insurance expert's tasks include assessing and estimating damage, evaluating the repairs to be made following an accident, reconstructing the responsibilities of the parties involved in a road accident, and conducting surveys when repairs have already been completed. They also have the responsibility to assess the extent of damage against individuals occurring in the course of an accident, trying to take into account their social and moral weight, in order to try to integrate this weight, as much as possible, within the compensation to be paid out.

This professional figure, as it acts as a super partes expert, has to ensure its independence and impartiality from the parties involved, meaning that it must have no ties to any of the parties involved in the outcome of their expertise. In fact, insurance experts cannot be an employee, nor an insurance mediator; they must have no relationships with insurance companies, nor work as a vehicle or boat repair mechanic for repair shops, or manage them. The expert is indeed considered by law as a freelancer who, as an expert, must assess the severity of the damage resulting from an accident and the necessary compensation. If a driver is found responsible, their insurance will be required to cover the damage caused to the other vehicle and individuals involved, so the expert role is crucial to determine the amount of compensation to be paid to the damaged party.

The intervention of an insurance expert is hence the single most important moment in settling damage compensation and must be arranged by the insurance company within

10 days of the accident to the company. Damage to individuals, on the other hand, must be assessed within 30 days of the delivery of medical healing certificates.

It is important to note that the amount of compensation depends on the responsible party's insurance coverage and the nature of the damage caused. In general, the insurance system in Italy is designed to ensure an efficient and prompt process of compensation for victims of road accidents.

Overall, the claims management process involves numerous individuals and necessitates considerable time with current systems. Additionally, insurance companies are frequently defrauded, as the affected individual may assert or manage to claim that damage predating the accident for which compensation is sought is caused by the accident itself, thereby obtaining a reimbursement exceeding the rightful amount. Another popular fraud involves the exchange of damaged parts among vehicles of the same model, leading to the same damage being reimbursed multiple times.

Figure 1 summarizes the presented claims management process and it highlights that the present claims management system is slow, occasionally inefficient, and demands significant human resources.

To address the aforementioned challenges, the objective of this study is to employ AI to support the processes of identifying vehicle damage and insurance reimbursement. This approach aims to significantly reduce evaluation times and enhance the accuracy of damage identification in accidents through computer vision and deep learning, providing support to the operation of insurance experts and minimizing their physical inspection of vehicles.

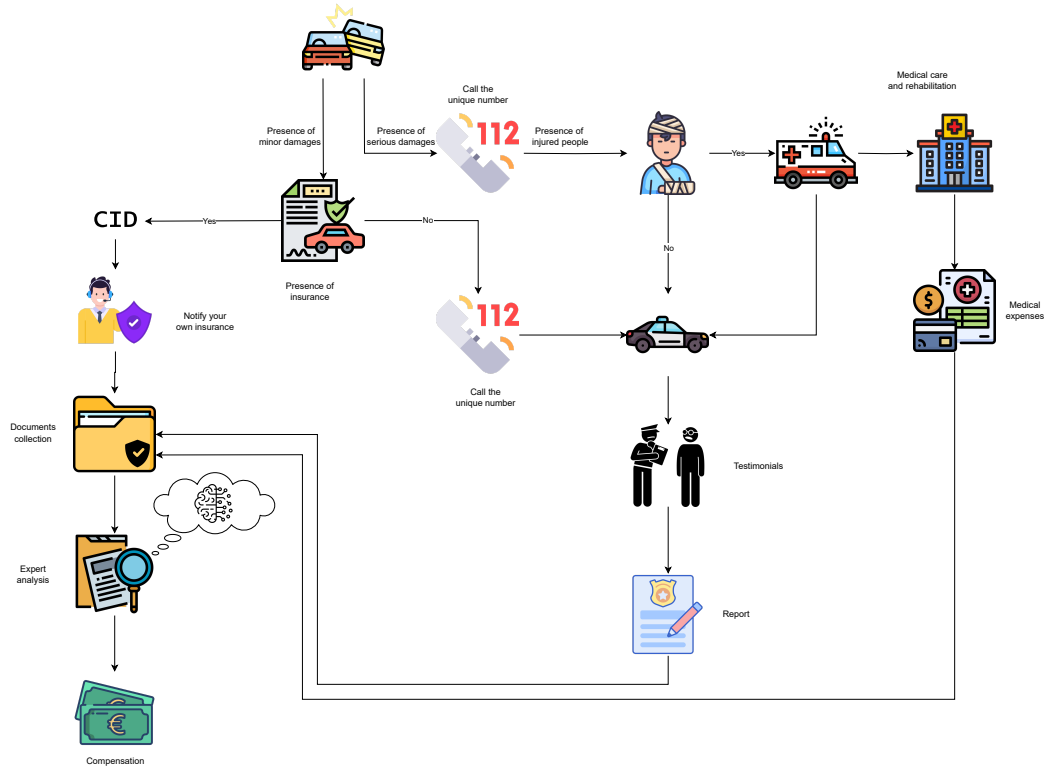


Figure 1. Block diagram of claims management.

Nevertheless, the proposed approach, tailored for the Italian case, is highly versatile and adaptable, making it easily extendable to European insurance companies. While it addresses the unique characteristics of the Italian context, it is based on universal principles and best practices. With appropriate adjustments for local specificities, this model can be successfully replicated across other European countries. The integration of advanced technological tools and AI ensures a solid foundation for the effective application on a continental scale.

3. Background of Computer Vision and AI for Vehicle Damage Detection

The issue of identifying and evaluating car damage is becoming increasingly significant in the literature. To date, existing approaches utilize a pipeline consisting of car detection, identification, and ultimately damage classification [3–5]. The literature follows two main approaches for identification:

- Detection [4], that is the process of recognizing damage automatically in photographs; this is used to select the most relevant images to be shown to a human expert to evaluate the damage further;
- Segmentation [6], that instead also localizes damage on a pixel level, providing a more precise damage identification at the cost of a more complex AI pipeline.

One of the most critical challenges in the development of an AI system capable of identifying damage is the difficulty in providing it with consistent and accurate training data, as estimating the severity of damage in photographs is a labour-intensive task and accident-related photographs are often confidential or contain private information such as registration plates [4,7]. Another technique related to this task's complexity is the detection of the pedestrian, which is performed by using infrared-converted temperature maps [8].

In fact, another common issue faced in the literature is the scarcity of data, as there is no publicly available dataset for this problem [4,9], making it hard to directly and fairly compare different approaches.

The target of the proposed system is to contribute to the automation of the damage estimation by providing a decision support system to insurance experts. For this reason, we split the damage identification procedure into three main steps, each of which will be tackled by a specific tool in our algorithmic pipeline:

(i) Damage detection;
(ii) Identification of damaged parts;
(iii) Estimation of repair costs.

The first stage is responsible for detecting and localizing car damage in pictures, independently from the photograph's angle. The second step involves the identification of which vehicle part is actually affected by the detected damage, independently from the car manufacturer and model. The final stage involves an accurate and timely estimation of repair costs, including market rates for labour and specific parts. The following subsection will discuss some related works and the criticality of each of the three subtasks.

The rationale behind this structure is to decompose a complex problem into smaller, more manageable components, allowing for easier solutions. In our specific process, the overarching issue is divided into three previously outlined steps. Rather than attempting to address the entire problem of estimating the repair costs associated with various instances of vehicle damage—taking into account both the type of damage and its location—we have categorized the problem into these three primary areas. This approach facilitates finding solutions more efficiently and effectively. By segmenting the problem, we can leverage targeted strategies for each category, enhancing both accuracy and efficiency in the repair cost estimation process.

The division of the process into three steps has also been used in other works and applications [10,11].

3.1. Damage Detection

Several studies exist in the literature on this subject. The first study we report is that of Patil et al. [9], who implemented fine-tuning from a pre-trained network on ImageNet. Fine-tuning is a machine learning technique based on using a network previously trained on a large and varied dataset to perform additional training, allowing the network to specialize in a specific task. In [9], damage is classified into dents, fractures, breaks, scratches, collisions, and the absence of damage. Classification is performed on the entire image using CNN-based classifiers, and damage localization is achieved by using classification on cropped images of the original image.

Among the most popular solutions for automated damage recognition, we mention Region-based Convolutional Neural Networks (RCNNs), which are convolutional neural networks [12] that work on image patches. RCNN algorithms are divided into two main categories: two-stage and one-stage. One-stage algorithms use each image only once to locate and classify the object, in this case, the damage. In contrast, two-stage algorithms work on image patches, performing recognition in two steps: identifying the regions where the object is likely to be and classifying the identified regions. Among the one-stage algorithms are the different versions of YOLO (You Only Look Once), which describe the idea of considering each image only once, while, for the two-stage algorithms, there are different versions of RCNN that indeed consider portions of images on which to perform damage detection.

Works such as [5] proposed solutions using YOLO, with the peculiarity of this work being to addresses antifraud measures, so re-identification features are extracted from damaged areas to allow the retrieval of damaged patches from an image database. Another significant work is represented by [6], where a damage detection segmentation algorithm based on transfer learning is proposed, together with a new Mask R-CNN with a custom semantic mask strategy.

3.2. Identification of Damaged Parts

A fundamental step for correctly estimating damage concerns the identification of which parts and components of a vehicle (e.g., bumper, doors) have actually been damaged.

Recent approaches tend to use a combination consisting of car identification, followed by damage assessment. These approaches become relevant when there is a need to automatically perform the task of damage assessment on images sent by users. Following this approach, a detection network is used to locate and crop the car from the larger image. Another network then identifies the class, position, severity, and size of the damaged area [4]. Alternatively, in [13], initially, the images entering the pipeline are filtered to select only exterior images of the car. Subsequently, the car view classification module classifies the sides of the car visible in the image, namely front, back, front-left, back-right, and so on. Based on this classification, enlarged images are sent directly to the damage recognition module, while other images are used to extract the brand and color of the car. Next, the vehicle view is classified using an EfficientNet-B3 [14] trained on nine classes (back, back-left, back-right, left, right, front, front-left, front-right, and enlarged). Data augmentation can be used to improve the performance of the classifier despite the low number of training images or, alternatively, transfer learning can be applied from a pre-trained network on COCO [6].

3.3. Estimation of Repair Costs

The analysis and assessment of damage in the context of insurance claims are increasingly becoming the focus of innovation and automation. Mallios et al. [15] present a comprehensive pipeline aiming to automate the damage assessment process, a crucial aspect for subsequent compensation by insurance companies. This approach is situated within a research landscape that encompasses various related studies. For instance, the approach by Imaam et al. [16] focuses solely on six car models, while Fernando et al. [17] integrate natural language processing to extract relevant information from claimants, proposing a complete pipeline for automating the insurance claim process. Another perspective is offered by Poon et al. [18], who present an end-to-end pipeline using Mask R-CNN to identify damaged areas, integrating metadata and image features to estimate costs. However, detailing damage descriptions based on specific car parts remain a challenge to be addressed. In this context, we thoroughly explore the proposals of these authors and identify the gaps that our research seeks to fill.

4. Proposed AI Pipeline

The contribution of our AI pipeline to the state of the art is twofold:

- The automatic estimation of damage severity in Insoore AI combines the knowledge extracted from the input images both in terms of damage size and damage type, whereas the solutions that are already existing in the market do not predict and use any insight in terms of damage type (for further details, see Section 4.2). As a result, the value added by the automatic assessment of damage type yields improved performance at estimating damage severity.
- To the best of the authors' knowledge, the performance achieved by the proposed AI pipeline at the combined tasks of (i) detecting damage and damaged parts and (ii) estimating damage severity [13]) on a heterogeneous test dataset assembled from footage coming from real-world road accidents, by relying on a Faster R-CNN architecture with fine-tuned aspect ratios (for further details, see Section 5.2).

In order to implement the three-step algorithmic pipeline, we first need to collect the vehicle images. For our purposes, we allow this collection to be carried out by either the insurance company's client or the expert, using a custom Android application that asks the user to collect pictures from a total of nine different angles.

The first activity we carry out is damage detection, with a deep neural network that classifies the detected damage appearing in the input image among four distinct classes (namely, broken clips, cracks, dents, and scratches).

Bodywork damage can be classified in various ways according to its severity. In the worst cases, an accident can lead to the destruction and deformation of a substantial part of the bodywork. In less severe cases, the damage may simply be limited to a scratch, dent, crack, or broken clip. This second category of damage is certainly the most widespread and the most easy to which to apply insurance fraud; for this reason, in this work, we narrow the attention to these categories of damage.

A crack on the car body refers to a fracture or break in the vehicle's surface, typically occurring in the paint, metal, or plastic components of the body. Cracks are more severe than scratches or dents and can compromise the structural integrity of the panel. Cracks typically require more advanced repair techniques, such as welding for metal parts or replacement of plastic panels, as well as refinishing with paint and a clear coat.

A dent on the car body is a deformation or depression in the vehicle's surface, usually caused by an impact or collision. Dents can vary in size and depth, ranging from small, shallow indentations to more significant and deeper depressions. They often occur due to minor accidents, such as door dings, hailstorms, or objects hitting the car. Depending on the severity, dents can affect the car's appearance and, in some cases, the structural integrity of the panel, though most minor dents are cosmetic in nature.

A scratch on the car body is a surface-level damage that typically results from an object scraping against the paint or clear coat. Scratches vary in depth and severity. Scratches can result from keys, tree branches, abrasive cleaning tools, or minor accidents. They typically affect the vehicle's appearance but can lead to rust or further damage if left untreated, especially deeper ones.

Broken clips on the car body refer to damaged fasteners that are used to secure various parts of the car's exterior or interior. These clips, typically made of plastic or metal, are essential for holding panels, trim, bumpers, and other components in place. When clips break, it can cause the attached parts to become loose, misaligned, or fall off.

For each detected damage instance, the model also outputs the coordinates of the damage within the image, together with a confidence score associated with the prediction, following an output structure analogous to that of the panel detector.

These four damage types represent the common types of damage that mainly occur. Focusing on these predominant damage types enables our AI system to efficiently detect and assess the most frequently occurring damage, enhancing the reliability of the claims process.

Our pipeline then proceeds with the segmentation of car parts (panels), which consists of categorizing each car part into one among fifty-three distinct classes. The fifty-three car panels encompass various sections of a vehicle.

- Front Section. This includes components such as the front bumper, grille, left-hand side fog light, right-hand side fog light, left-hand side headlight, right-hand side headlight, bonnet, and front windscreen.
- Side Sections. These cover parts like the left-hand side view mirror, right-hand side view mirror, left fender, right fender, left front door, right front door, left rear door, right rear door, left front wheel, right front wheel, left rear wheel, right rear wheel, left front window, right front window, left rear window, right rear window, left quarter panel, right quarter panel, left bottom rail, right bottom rail, left top rail, and right top rail.
- Rear Section. This consists of the boot (dickey), rear windscreen, rear bumper, left-hand side tail light, right-hand side tail light, centre tail light, and spoiler.
- Additional Components. These include the roof, licence plate, sensors, door handles, caps, reflectors, indicators, and other miscellaneous parts.

By categorizing the vehicle into these specific panels, our AI system can more precisely identify and assess damage, leading to more accurate claim processing.

In Figure 2, the main panels are detected and segmented. For each detected panel, the model provides three outputs: a segmentation map for the various detected parts, the corresponding panel type, and the inference confidence score associated with the prediction.



Figure 2. Panel recognition results for multiple car models from various different angles.

Combining the outputs from both the panel detector and the damage detector, damage is mapped to its corresponding panels, and the relative area of the damage with respect to its panel is calculated. Based on this relative area and the type of damage, the damage severity is classified among three possible levels: low, medium, and high severity.

Finally, the pipeline returns an aggregated severity assessment for each damaged panel, determining whether it should be repaired or replaced, based on the severity and damage type of each single damage appearing on the same panel and a list of part and labour prices updated by the insurance company daily. Figure 3 provides a high-level visualization of the proposed functional architecture.

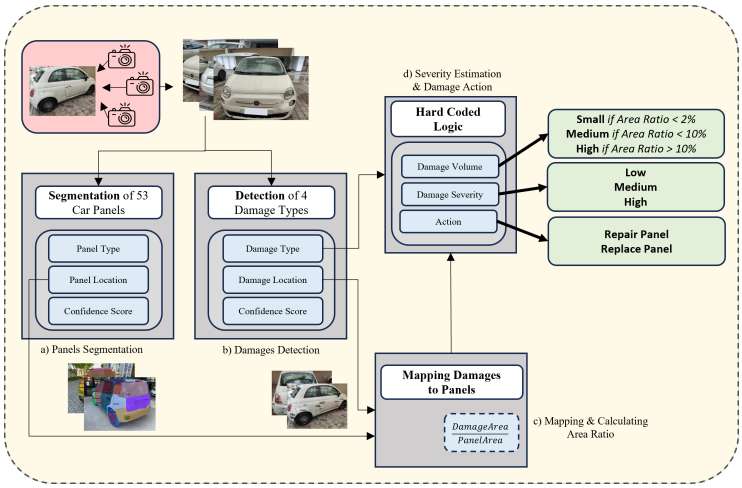


Figure 3. Flowchart of the proposed AI system for automatic damage detection.

4.1. Automatic Identification of Damage and Damaged Parts

Several experiments were conducted to obtain the reported values. The metrics used to evaluate the performance of the models included average precision (AP), and average precision at Intersection over Union (IoU) thresholds of 50% (AP50) and 75% (AP75). Among these, AP50 was considered the primary metric for assessment. Definitions for each metric are provided in Appendix A. Thirteen experiments were performed, based on the Faster R-CNN architecture, a widely adopted approach in recent years for object detection. Figure 4 reports some examples of recognized damage, whereas Figure 2 shows the results of panel identification for several different car models.

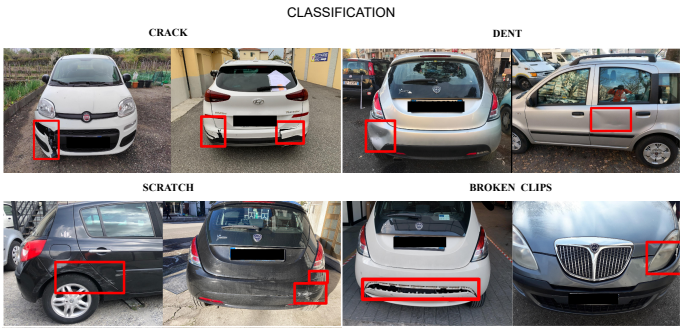


Figure 4. Damage classification: results obtained from our custom classifier for the four considered classes.

4.2. Automatic Estimation of Damage Severity

The severity estimation algorithm is designed to assess the extent of damage on vehicle panels and to recommend appropriate actions—either repair or replacement—based on the severity of the damage. The algorithm pseudocode in Algorithm 1 follows these key steps:

- Initialization: An empty list, *damaged_panels*, is created to store all detected damaged panels.
- Damage mapping: For each detected damage instance in the input list *damages*, the algorithm checks whether the corresponding panel is already present in *damaged_panels*. If the panel is not present, a new panel element is created. This element includes a map of the damage and a list containing the damage and its classification. The newly created panel is then added to *damaged_panels*, and the area of the panel is determined based on the panel mask. If the panel is already present in *damaged_panels*, the existing damage map is updated with the new damage and its classification.
- Action and severity assessment: An empty list, *damaged_panels_and_action*, is created to store the damaged panel, the overall volume of damage, the severity estimation, and the recommended action. For each panel in *damaged_panels*, the initial action is set

to 'repair' and the severity to 'low'. If the classification of the damage on the panel matches any of the entries in *replace_ids*, the action is updated to 'replace' and the severity to 'high'. The area of the damage map is calculated and used to determine the percentage of damage relative to the total panel area (denoted as *damage_vol*). If the initial action is 'repair' and *damage_vol* exceeds 0.4, the action is changed to 'replace' and the severity to 'high'. If *damage_vol* is greater than 0.1 but does not exceed 0.4, the severity is updated to 'medium'.

- Result compilation: Each panel, along with its severity, recommended action, and damage volume, is appended to *damaged_panels_and_action*.
- Output: The final list, *damaged_panels_and_action*, is returned, providing a comprehensive severity assessment for each damaged panel, including the severity of the single damage instance and the recommended action to be taken. This systematic approach ensures that each detected damage is accurately mapped, evaluated in terms of severity, and assigned an appropriate remediation action, thus facilitating efficient and effective damage management in vehicle inspection processes.

Algorithm 1 Severity Estimation Pseudocode

Require: *image_size, damages, repair_ids, replace_ids*
Create an empty list *damaged_panels* to insert all damaged panels
1: Create an empty list *damaged_panels* to insert all damaged panels
2: for all damage $\in$ damages **do**
3: **if** *Panel(damage)* $\notin$ *damaged_panels* **then**
4: Create a panel element containing a map of damage and a list containing damage and its classification
5: Add the created panel to the list *damaged_panels*
6: Get the panel area from the mask
7: **else**
8: Update the map of damage of *Panel(damage)* in *damaged_panels*
9: Add the damage and its classification to the list of the damage of the given panel
10: end if
11: end for
12: Create an empty list *damaged_panels_and_action* to insert the damaged panel, the volume of the overall damage, the severity estimation, and the action to perform
13: for all panel $\in$ *damaged_panels* **do**
14: *Action* $\leftarrow$ *'repair'*
15: *Severity* $\leftarrow$ *'low'*
16: **if** classification of damage $\in$ *replace_ids* **then**
17: *Action* $\leftarrow$ *'replace'*
18: *Severity* $\leftarrow$ *'high'*
19: end if
20: Calculate area of damage map $\rightarrow$ *damage_area*
21: Calculate the percentage of damage $\rightarrow$ *damage_vol*
22: **if** *Action* $==$ *'repair'* **then**
23: **if** *damage_vol* > 0.4 **then**
24: *Action* $\leftarrow$ *'replace'*
25: *Severity* $\leftarrow$ *'high'*
26: **else if** *damage_vol* > 0.1 **then**
27: *Severity* $\leftarrow$ *'medium'*
28: **end if**
29: end if
30: Append *(panel_id, Severity, Action, damage_vol)* to *damaged_panels_and_action*
31: end for
32: return *damaged_panels_and_action*

5. Results

5.1. Dataset

Hereinafter we report the results obtained by the damage detector in the proposed algorithm pipeline, which to the best of the authors' knowledge is the stage where most of the complexity lies.

The data used for our tests consisted of a private dataset containing photos of vehicle damage coming from real-world road accidents. In particular, in line with our design, the dataset was labelled according to four damage classes: cracks, dents, scratches, and broken clips.

In the process of annotating vehicle damage in images, we adhered to specific guidelines for consistency and accuracy. First, we excluded undamaged images and images depicting disassembled vehicles located at a mechanic's workshop. For damage affecting multiple body parts, we used a single annotation to encompass all affected areas. Different types of damage, even if occurring in the same area of a car panel, were annotated separately. If two damage instances, such as dents, were in close proximity, we annotated them as a single instance; however, if the damage instances were small but occurred in different parts of the panel, we annotated them separately. Multiple 'scratch' swipes were grouped into a minimal number of annotations. Dent annotations were aligned approximately with the shadowed area created by the damage. The 'crack' class required a single annotation that included the entirety of the split or break in the component. For the 'broken clips' class, the annotation always enclosed the black gap resulting from the breakage of the hooks holding two body parts together.

The samples per dataset as shown in Table 1 are arranged as follows: (1) The training set is composed of 21,846 total annotations from 6782 total images. (2) The test set is composed of 540 annotations from 326 images. The dataset was annotated following the COCO dataset format [19].

Table 1. Sample distribution per class for the training and test sets.

Data	Scratch	Dent	Crack	Broken Clips	Annotations
training	47.64%	23.89%	16.40%	12.07%	21,846
test	46%	33%	11%	10%	540

The class distribution in the dataset reflects the realistic conditions of a road accident scenario. In particular, we extracted around 120,000 images alongside with business attributes of interest such as the monetary value of the damage, the brand, and the model of the car, etc.; then, we mimicked the same distribution of those attributes on the test set ending up having a sample of the test set that is identical to the real-world distribution of these business features of interest.

5.2. Experiments and Performance Assessment

Several experiments were conducted. The performance evaluation metrics of the model were as follows:

- AP (average precision);
- AP50 (average precision for IoU greater than 50%);
- AP75 (average precision for IoU greater than 75%).

Among these, the main metric we observed was AP50.

The first two experiments were a first step towards reproducing Mask R-CNN as suggested in a previous paper on damage detection on vehicles in an insurance claim context [13]. Experiments 1 and 2 in Table 2 are the result of such an experiment in our dataset. All experiments onward are based on the Faster R-CNN architecture, which is an approach widely used in recent years for object detection [20].

Table 2. Fine- tuning experiment results.

N	Fine-Tuning Experiments	AP	AP50	AP75
1	Mask R-CNN (Seg)	1.02	4.89	0.14
2	Mask R-CNN (BB)	11.92	28.16	8.2
3	Default: Faster R-CNN + FPN:X101	11.24	28.16	7.2
4	Data Aug.1	8	20.55	5.2
5	Data Aug.2	6.7	18.27	3.87
6	Backbone1	12.4	28.68	9.09
7	Backbone2	11.93	28.22	8.47
8	FPN1	8.86	24.34	4.22
9	FPN2	6.3	18.04	2.21
10	RPN: Anchors Size1	12.38	29.52	6.62
11	RPN: Anchors Size2	12.31	30.45	7.29
12	RPN: Anchors Size3	12.15	29.05	7.72
13	RPN:AnchorsAspect Ratio1	12.65	29.28	9.63
14	RPN:AnchorsAspect Ratio2	12.41	30.04	8.21
15	RPN: Hybrid1	11	26.71	8.52

The initial experiment on Faster R-CNN was established as the baseline for comparison. Subsequent experiments introduced a singular modification relative to the baseline configuration to precisely observe the impact of each alteration on performance metrics. The configuration for Experiment 3 employed the default settings from Detectron2 [21]: Faster R-CNN, utilizing faster_rcnn_X_101_32x8d_FPN_3x as the backbone.

In Experiment 4, the data augmentation hyperparameters were altered, changing the minimum size from (400, 500, 600, 700, 800, 900, 1000, 1100, 1200) to 1300 and the maximum size from 4000 to 4096 for both training and testing. Experiment 5 similarly modified the minimum and maximum size data augmentation hyperparameters, adjusting them from the default values to be sampled from the list (640, 672, 704, 736, 768, 800, 1300). In Experiment 6, the number of frozen (non-trained) stages was adjusted from the default value of 2 to 0. Similarly, Experiment 7 modified the default number of frozen stages from 2 to 1. In Experiment 8, the output channel of the Feature Pyramid Network (FPN) was altered from the default setting of 256 to 128. Experiment 9 modified the FPN's output channel from the default 256 to 512. Experiments 10, 11, and 12 involved changes to the default settings of the anchor generator sizes from [[32, 64, 128, 256, 512]] to [[4, 8, 16, 32, 64, 128, 256, 512]], [[8, 16, 32, 64, 128, 256, 512]], and [[16, 32, 64], [32, 64, 128], [64, 128, 256], [128, 256, 512], [256, 512, 1024]], respectively. In Experiment 13, only the aspect ratios were modified from the default values of [[0.5, 1.0, 2.0]] to [[0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0]]. Experiment 14 similarly adjusted the aspect ratios from the default [[0.5, 1.0, 2.0]] to [[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1.0, 2.0]]. Experiment 15 introduced two changes relative to the baseline: both the anchor generator sizes and aspect ratios were adjusted from the default sizes of [[32, 64, 128, 256, 512]] to [[8, 16, 32, 64, 128, 256, 512]] and from the default aspect ratios of [[0.5, 1.0, 2.0]] to [[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1.0, 2.0]].

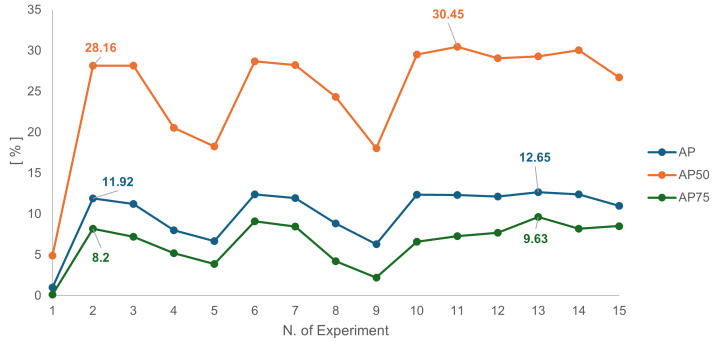


Figure 5. Performance curve of the experiments carried out.

5.3. Discussion of the Experimental Results

According to the results in Table 2 and Figure 5, we found that the anchors' size and aspect ratio are the key factors for achieving better performance. In particular, in Experiments 11, 13, and 14, the following modifications were made to the Faster R-CNN network's anchor generator settings.

- Experiment 11. The anchor generator sizes were changed from the default configuration of [[32, 64, 128, 256, 512]] to [[8, 16, 32, 64, 128, 256, 512]]. By introducing smaller anchor sizes, this modification enabled the network to generate anchors that better captured smaller objects in the input images. The goal was to improve detection performance, particularly for objects at lower spatial scales.
- Experiment 13. In this experiment, the focus was on adjusting the aspect ratios. The default aspect ratios of [[0.5, 1.0, 2.0]] were modified to a more varied set: [[0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0]]. This allowed the Faster R-CNN network to handle objects of diverse shapes more effectively by providing a wider range of anchor aspect ratios, which can accommodate objects with more nuanced height-to-width relationships.
- Experiment 14. In this experiment, the aspect ratios were further expanded from the default [[0.5, 1.0, 2.0]] values to [[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1.0, 2.0]]. This fine-tuning of aspect ratios introduced a higher resolution for detecting objects with extreme aspect ratios, such as very tall or very wide objects, enabling the network to perform more robustly across a broader range of object geometries.

In summary, Experiment 11 introduced smaller anchor sizes to improve small-object detection, while Experiments 13 and 14 focused on expanding the range of aspect ratios to capture objects with a wider variety of shapes more accurately.

We demonstrated that real-life data, such as ours, is significantly more challenging than CarDD [22], as evidenced by the comparatively lower performance metrics we obtained. Object segmentation is notably more difficult than object detection as shown in the table. In particular, our results indicate that, with 6782 images, Mask R-CNN [23] networks struggle to learn masks, whereas its detection network performs reasonably well. Therefore, we conclude that object detection is a more suitable solution for this problem, as opposed to object segmentation, which has been widely attempted in other studies, such as 'A deep-learning–based antifraud system for car-insurance claims' [13]. Consequently, we adopted object detection and fine-tuned our model, surpassing both the Mask R-CNN detection network and the Faster R-CNN baseline experiment. The best overall performance was achieved in Experiment number 13, while the highest AP50 was observed in Experiment 11.

6. Conclusions and Future Works

This paper reported on the insurtech solution Insoore AI, which integrates deep neural networks for computer vision to automate damage identification and localization for claims management with a specific focus on the Italian market. The proposed solution is being integrated into a decision support system with the aim of assisting insurance experts in correctly estimating repair costs. Future work will be aimed at improving the damage detection performance using vision transformers and at automating the estimation of repair costs.

Author Contributions: Conceptualization, M.M.H.A., V.B., A.G., V.A., G.G. (Gerardo Gorga), F.S. and E.S.; methodology, A.G. and A.P.; software, M.M.H.A. and V.B.; validation, G.G. (Gonzalo Gutierrez), A.O., M.A.R. and L.R.C.; formal analysis, G.G. (Gonzalo Gutierrez), A.O., M.A.R. and L.R.C.; investigation, G.G. (Gonzalo Gutierrez), A.O., M.A.R. and L.R.C.; resources, G.G. (Gonzalo Gutierrez), A.O., M.A.R. and L.R.C.; data curation, G.G. (Gonzalo Gutierrez), A.O., M.A.R. and L.R.C.; writing—original draft preparation, M.M.H.A. and V.B.; writing—review and editing, M.M.H.A. and V.B.; visualization, G.G. (Gonzalo Gutierrez), A.O., M.A.R. and L.R.C.; supervision, A.G., A.P., V.A., G.G. (Gerardo Gorga), F.S. and E.S.; project administration, A.G., A.P., M.P. and L.R.C.; funding acquisition, A.G., A.P. and L.R.C. A.O. is currently an independent researcher; yet, the work presented

in this paper has been carried out in the context of A.O.'s activity at Whoosnap until October 2024. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Insoore AI project, POR FESR LAZIO 2021–2027—Avviso Pubblico Riposizionamento Competitivo RSI—Ambito Industrie creative e digitali, co-funded by Regione Lazio and the European Union, under Grant CUP F89J23000730007.

Data Availability Statement: Data is unavailable due to privacy restrictions.

Acknowledgments: The authors thank the software development team at Insoore for the support provided in terms of data engineering and platform implementation.

Conflicts of Interest: We report that E. Scianaro, V. Arconzo and G. Gorga are founders of Whoosnap Srl, F. Santini is executive chairman of Whoosnap Srl, and G. Gutierrez, M. A. Rangisetty and L. Ricciardi Celsi are employees at Whoosnap Srl.

Appendix A. Evaluation Metrics

As detailed in [24], the COCO metrics, initially introduced in conjunction with the COCO dataset, have become the preferred evaluation method for models in object detection and segmentation mapping. All experiments were evaluated using COCO metrics, first proposed in Microsoft COCO: Common Objects in Context [19] and further listed on Microsoft COCO Website [25]. According to the specifications provided by Microsoft COCO, the terms average precision (AP) and mean average precision (mAP) are used interchangeably.

In [24], the average precision is well defined used in COCO metrics: Precision is the ratio of true positives to the sum of true positives and false positives and it is calculated using the standard precision formula; this approach is not particularly effective in the context of object detection, where models may produce numerous predictions, some with low confidence scores. Average precision (AP) addresses this limitation and is recognized as a robust general-purpose metric for evaluating object detection models.

As defined in [24], Intersection over Union, or IoU, is an area-based metric used in Computer Vision to determine if two objects can be considered to be the same. As the name suggests, the metric is computed by dividing the area of the intersection of two objects by the area of their union. The range of IoU is 0 to 1, with two identical objects having an IoU of 1 and objects with no intersection possessing an IoU of 0.

AP: This is the average precision at ten IoU thresholds ranging from IoU = 0.50 to IoU = 0.95 (IoU = 0.50: 0.05: 0.95) [25].

AP50: This is the average precision calculated at an IoU threshold of 50% [25]. AP75: This is the average precision calculated with an IoU threshold of 75% [25].

References

1. Andreozzi, A.; Ricciardi Celsi, L.; Martini, A. Enabling the Digitalization of Claim Management in the Insurance Value Chain Through AI-Based Prototypes: The ELIS Innovation Hub Approach. In *Digitalization Cases*; Urbach, N., Röglinger, M., Kautz, K., Alias, R.A., Saunders, C., Wiener, M., Eds.; Management for Professionals; Springer: Cham, Switzerland, 2021; Volume 2. [CrossRef]
2. Legge 17 febbraio 1992, n. 166. *GazzettaUfficiale della Repubblica Italiana*. 1992. Available online: https://www.gazzettaufficiale.it/eli/id/1992/02/27/092G0218/sg (accessed on 28 June 2024).
3. Bandi, H.; Joshi, S.; Bhagat, S.; Deshpande, A. Assessing car damage with convolutional neural networks. In Proceedings of the 2021 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India, 25–27 June 2021; pp. 1–5.
4. De Deijn, J. Automatic car damage recognition using convolutional neural networks. In Proceedings of the 2018 Internship Report MSc Business Analytics, Amsterdam, The Netherlands, 1 December 2018.
5. Li, P.; Shen, B.; Dong, W. An anti-fraud system for car insurance claim based on visual evidence. *arXiv* **2018**, arXiv:1804.11207.
6. Zhang, Q.; Chang, X.; Bian, S. Vehicle-damage-detection segmentation algorithm based on improved mask RCNN. *IEEE Access* **2020**, *8*, 6997–7004. [CrossRef]
7. Van Ruitenbeek, R.; Bhulai, S. Convolutional Neural Networks for vehicle damage detection. *Mach. Learn. Appl.* **2022**, *9*, 100332. [CrossRef]

8. Zhao, Y.; Cheng, J.; Zhou, W.; Zhang, C.; Pan, X. Infrared pedestrian detection with converted temperature map. In Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 18–21 November 2019; pp. 2025–2031.
9. Patil, K.; Kulkarni, M.; Sriraman, A.; Karande, S. Deep learning based car damage classification. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 50–54.
10. Tukra, S.; Hoffman, F.; Chatfield, K. Improving visual representation learning through perceptual understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14486–14495.
11. Bdeo. Visual Intelligence for Motor and Home Insurance Companies. 2024. Available online: https://bdeo.io/en/ (accessed on 16 October 2024).
12. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
13. Maiano, L.; Montuschi, A.; Caserio, M.; Ferri, E.; Kieffer, F.; Germanò, C.; Baiocco, L.; Celsi, L.R.; Amerini, I.; Anagnostopoulos, A. A deep-learning—Based antifraud system for car-insurance claims. *Expert Syst. Appl.* **2023**, *231*, 120644. [CrossRef]
14. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
15. Mallios, D.; Li, X.; McLaughlin, N.; Del Rincon, J.M.; Galbraith, C.; Garland, R. Vehicle damage severity estimation for insurance operations using in-the-wild mobile images. *IEEE Access* **2023**, *11*, 78644–78655. [CrossRef]
16. Imaam, F.; Subasinghe, A.; Kasthuriarachchi, H.; Fernando, S.; Haddela, P.; Pemadasa, N. Moderate automobile accident claim process automation using machine learning. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; pp. 1–6.
17. Fernando, N.; Kumarage, A.; Thiyaganathan, V.; Hillary, R.; Abeywardhana, L. Automated vehicle insurance claims processing using computer vision, natural language processing. In Proceedings of the 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 30 November–1 December 2022; pp. 124–129.
18. Poon, F.; Zhang, Y.; Roach, J.; Josephs, D.; Santerre, J. Modeling and Application of Neural Networks for Automotive Damage Appraisals. *SMU Data Sci. Rev.* **2021**, *5*, 3.
19. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part V 13. Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef] [PubMed]
21. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on October 22nd, 2024).
22. Wang, X.; Li, W.; Wu, Z. CarDD: A New Dataset for Vision-based Car Damage Detection. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7202–7214. [CrossRef]
23. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
24. Wood, L.; Chollet, F. Efficient Graph-Friendly COCO Metric Computation for Train-Time Model Evaluation. *arXiv* **2022**, arXiv:2207.12120. Available online: https://arxiv.org/abs/2207.12120 (accessed on 22 October 2024).
25. Microsoft-COCO. Microsoft COCO: Common Objects in Context. Available online: https://cocodataset.org/#detection-eval (accessed on 22 October 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.