Expert Systems With Applications 231 (2023) 120644

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A deep-learning–based antifraud system for car-insurance claims

Luca Maiano [a,b,*], Antonio Montuschi [c], Marta Caserio [c], Egon Ferri [b], Federico Kieffer [b], Chiara Germanò [b], Lorenzo Baiocco [b], Lorenzo Ricciardi Celsi [a,b], Irene Amerini [a], Aris Anagnostopoulos [a]

[a] Sapienza University of Rome, via Ariosto, 25, 00185 Rome, Italy
[b] ELIS Innovation Hub, Via Sandro Sandri, 81, 00159 Rome, Italy
[c] Assicurazioni Generali, Piazza Tre Torri, 120145 Milan, Italy

ARTICLE INFO

Keywords:
Image similarity
Image search
Fraud detection
Car insurance

ABSTRACT

The annual cost of vehicle insurance fraud is estimated to exceed 40 billion dollars. This is an enormous amount considering the number of new vehicles insured yearly. In terms of higher premiums, it implies that insurance fraud incurs an additional annual cost to each U.S. family of $400 to $700, on average. Many frauds can be attributed to previously reported damages, which are submitted a second time to the insurance company. In these cases, it does not suffice to check the customer's history to identify them: Damaged car panels can be removed from one vehicle and reassembled on another to make an insurance claim on the second car. To deal with these fraud attempts, in this paper, we propose an end-to-end solution to support the *special investigation unit* of the insurance companies in their antifraud investigations. For each claim, we organize the images sent to the insurance company and analyze them to extract basic vehicle information. Subsequently, we use these images to identify any damage to the bodywork, and, finally, we verify that the damage has not already been processed in previous claims. To the best of our knowledge, this is the first published work that deals with this problem through a pipeline that covers the entire claim management process. To validate our proposal, we compare our solution with other state-of-the-art models for estimating image similarity. Our results show that our solution is, on average, superior by 15.27% in terms of mean average precision (mAP). In addition, we report the challenges faced in scaling such a system in a production environment. This aspect is often ignored, but applying these solutions to industrial settings is of fundamental importance. Our proposed end-to-end system can reduce by up to 18% the number of false positives produced by the damage reidentification system. We show that encapsulating several specialized components and merging their intermediate results leads to a 72% reduction in possible alerts. Finally, to support the discussion and comparison of explanations for this new task, we introduce a new dataset as a benchmark for damage reidentification.

1. Introduction

The insurance industry consists of thousands of companies, which collect over one trillion dollars in premiums every year. The massive size of the industry contributes significantly to the cost of insurance fraud by providing more opportunities and more incentives to commit illegal activities. Indeed, the annual cost of vehicle insurance fraud is estimated to exceed 40 billion dollars (FBI, 2022). Added to this, insurance fraud is often used to fund the wider activities of criminal gangs, which may be linked to serious organized crime such as drug dealing, burglary, or terrorism (Association of British Insurers (ABI), 2022). For this reason, insurance companies have developed processes to detect, disrupt, and prosecute people who try to fabricate a claim. Advanced analytics software helps insurers proactively identify cross-industry patterns and alert the industry to fraudulent networks. This work is in this direction: We introduce an end-to-end pipeline designed to detect automotive damage fraud.

Insurance companies process a very large amount of images every day. Customers who make a claim for car damage are required to upload several photographs of the involved vehicle, which allow the insurance company to examine the damage as well as the vehicle as a whole. These images include photos of the exterior or interior of

* Corresponding author at: Sapienza University of Rome, via Ariosto, 25, 00185 Rome, Italy.
E-mail addresses: maiano@diag.uniroma1.it (L. Maiano), antonio.montuschi@generali.com (A. Montuschi), marta.caserio@generali.com (M. Caserio), e.ferri@elis.org (E. Ferri), f.kieffer@elis.org (F. Kieffer), c.germano@elis.org (C. Germanò), l.baiocco@elis.org (L. Baiocco), ricciardicelsi@diag.uniroma1.it (L. Ricciardi Celsi), amerini@diag.uniroma1.it (I. Amerini), aris@diag.uniroma1.it (A. Anagnostopoulos).

https://doi.org/10.1016/j.eswa.2023.120644
Received 30 August 2022; Received in revised form 23 February 2023; Accepted 29 May 2023
Available online 2 June 2023

Fig. 1. Example of damage similarities. Triplets (a)–(b)–(c), (d)–(e)–(f), and (g)–(h)–(i) are real matches of the same damages. Each row shows an example type of damage: *scratch*, *crack*, and *dent*, respectively.

the vehicle, the insured's documents, the vehicle registration document, details of the damage, the license plate, and the car's vehicle identification number (VIN). These images flow into the claim management process, through which the insurance experts manually inspect the correspondence between the claim reported and the information present in the images. This process requires extracting a significant number of information from the images, such as the correspondence of the vehicle in the image with the insured one, the verification of the license plate number (Cantarini et al., 2020; Djara et al., 2017; Etomi & Onyishi, 2021; Jaderberg et al., 2016; Jain et al., 2016; Lubna et al., 2021; Manana et al., 2021; Yaacob et al., 2021), the VIN, the color of the car, or the presence of damages on the bodywork (Bandi et al., 2021; Kyu & Woraratpanya, 2020; Malik et al., 2020; Patil et al., 2017). Extracting manually all these data requires a very large amount of time and effort and has a significant effect on the costs incurred by the insurance companies. Added to this is the need to deal with increasingly sophisticated attempts at fraud (Association of British Insurers (ABI), 2022; FBI, 2022). In some cases, previously reported damages are reproposed to the same insurance company to obtain new compensation. A first idea to address this problem could be to verify that the vehicle analyzed has not suffered the exact same damage in the past; yet this is not always sufficient: In fact, in the most sophisticated cases, the damaged bodywork component is removed from the vehicle and reassembled on another car! Thus, to identify this type of fraud, requires to inspect the damages and compare damages among different vehicles. Of course, the adversary can even change the damage by scratching more or hitting the already damaged part, which makes this problem even harder to solve. Identifying these cases among the millions of images processed every year is an extremely complex task to automate because of the enormous heterogeneity of the collected data. Different damages of the same type can have different shapes, sizes, and colors. Added to this, reflections or dirt on the bodywork can make it even more difficult to identify them. In this paper, we introduce a new pipeline, which is designed to support the experts to automatically identifying possible fraud attempts. Fig. 1 shows some examples of this problem for three types of damage.

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*, or *crack*. This second category of damages is certainly the most widespread and the most easy to apply insurance fraud; for this reason in this work we narrow the attention to these categories of damage. However, recognizing them can be very complex. Each damage can have a very different shape and size from any other. An additional complication arises from the necessity to be able to recognize these damages in spite of reflections, light conditions, partial occlusions, zoom-level, blurring, or dirt on the bodywork. The problem becomes even harder because of the need to find the same damage among millions of images, in which (even worse) the photograph of the damage may have been taken from a different angle and under different environmental conditions (lighting, background, etc.). Unfortunately, unlike other tasks such as person reidentification (which we discuss in Section 2.2), this problem has been little addressed on cars because of the scarce availability of open data available to explore new possible solutions.

The main contributions of this paper can be summarized as follows:

- We introduce a new benchmark dataset for the recognition of similar damages; with this, we hope to stimulate discussion on this type of problem and to make a common dataset available to the community to evaluate the proposed solutions.
- We propose an end-to-end pipeline for *damage similarity* detection. As far as we know, this is the first work that proposes to

investigate the possibility of recognizing this type of fraud with a pipeline that manages the entire process from image acquisition to signaling of possible similar damage.
- We discuss the difficulties encountered in scaling these solutions in a real-world setting.

In detail, our proposed system is structured in the following different phases: images sent by policyholders are initially filtered to select only those containing the exterior of the vehicle. The car is then detected within the image. At this point, the system classifies the color and brand of the vehicle and localizes the damages present on the car. Finally, the identified damages are mapped within an embedding and compared with those of the claims already analyzed in the past, filtering the possible matches with respect to the color, brand and view of the vehicle. This filtering is intended to reduce the number of comparisons to be made and reduce the possible number of incorrect matches.

The rest of this paper is organized as follows. In Section 2, we begin by reviewing the actual state of the art solutions and comparing these methods with our proposed pipeline. Section 3 describes the proposed pipeline and Section 4 discusses the implementation details with an in-depth look at the dataset and the evaluation metrics. Finally, in Sections Section 5, 6, and 7 we report our experiments and we draw some discussion before concluding the paper.

2. Related work

Insurance companies receive thousands of new claims every day, each containing several images. After being taken over, insurance company experts must analyze all the images of a claim to decide how to conclude the compensation process. For the larger insurance companies, this translates into having to analyze millions of images every year. However, managing such a large amount of data is extremely expensive in terms of human resources and cost of maintaining these processes. Furthermore, detecting fraud attempts on millions of claims is an even more complex task. As a result, many insurance companies have started developing image-analysis solutions to automate part of the claim management process. In the following sections we focus on the fundamental blocks of the pipeline proposed in this work: (1) damage recognition systems and (2) deep learning techniques for object reidentification in images.

2.1. Damage detection

The enormous heterogeneity of damages and the lack of large labeled datasets makes it difficult to train robust damage classifiers. In addition to this, being a very different task compared to traditional object detection tasks in which a certain object to be identified has a more or less homogeneous shape, it is not obvious that using pretrained models can improve the performance of a damage classifier. In sight of this (Patil et al., 2017) consider a wide range of damages such as dent, glass shatter, broken lamp, scratch or smash, and propose a series of experiments in which they compare the effectiveness of different approaches including (1) training a CNN, (2) unsupervised pretraining of an auto-encoder followed by a fine-tuning, (3) using of transfer learning from CNN trained on ImageNet (Deng et al., 2009) and (4) creating an ensemble classifier on top of the set of pretrained classifiers. A similar approach is proposed by Kyu and Woraratpanya (2020), who collect a dataset of damaged and undamaged car images from the web and fine-tune a pretrained VGG (Simonyan & Zisserman, 2014) with L2 regularization to contain overfitting. The results from Patil et al. show that transfer learning combined with ensemble learning works best. However, ensemble learning can be computationally expensive, leading to the increase of maintenance cost of an automated claim management pipeline. On the contrary, in this paper, we propose a simpler approach that allows to obtain acceptable performance without requiring too many computational resources. With the same hardware available, this allows you to optimize the claim management process without increasing processing times or costs per image.

Accurately recognizing damages is not enough to automate the entire damage detection process. To use these methods in production it is first of all necessary to select only the images that may contain the damage. For this reason, in many cases, damage detection models are often preceded by other models that deal with filtering images containing vehicles and are used in parallel with another car-panel detection system that allows damage to be localized on the bodywork. In this direction, Bandi et al. (2021) propose an approach based on a pipeline made up of four models: (1) a filter that discards images that do not contain cars, (2) a classifier that identifies damages to the bodywork, and two parallel classifier estimating (3) the severity of the damage, and (4) position (side, rear, front). Khan et al. (2021) propose a similar methodology. In terms of deep-learning architectures, the Mask R-CNN (He et al., 2018) is a common and accurate solution for damage and panels detection (Zhang et al., 2020; Zhu et al., 2019). Zhu et al. (2019), propose a pipeline consisting of a Mask R-CNN for identifying vehicle panels, a RetinaNet (Lin et al., 2018) used for damage recognition and an Inception-V3 (Szegedy et al., 2015) network that classifies the type of damage and the corresponding severity. Differently, in our work, the localization of the damage is obtained by classifying the view of the vehicle, that is back, front, left, right, back-left, and so on. However, the complexity of car-damage detection and segmentation may lead to lower detection segmentation accuracy and slower detection speed. Therefore, Zhang et al. (2020) propose a modification of the ResNet-50 (He et al., 2015) network. By reducing the number of layers in the residual network, and adjusting the internal structure to strengthen the regularization of the model, they enhance its generalization ability. Compared to these works, in this paper, we choose to adopt the Mask RCNN for damage detection and to filter possible matches based on the view of the vehicle. Through our experiments, we show that this model performs really well in a real setting. Moreover, our pipeline is based on a filtering step that selects images containing vehicles, a vehicle detection module that retrieves the position of the vehicle in the image, and brand and color detection systems that extract the information about the car. All these modules are in handy to produce an end-to-end damage detection system.

2.2. Deep-reidentification architectures

The lack of data and understanding of the challenges associated with insurance fraud by people outside the insurance business has not attracted the scientific community's interest in these problems. To our knowledge, the only work dealing with damage reidentification has been proposed by Li et al. (2018). The paper proposes to use the YOLO (Redmon et al., 2016) network as a local damage detector and a pretrained VGG16 as a global feature descriptor. By fusing the features extracted by the last convolutional layer of the VGG16 with a color histogram, they obtain a more discriminative global descriptor. The local and global descriptors are finally concatenated and compared with an image history via the cosine distance. Differently from Li et al. (2018), in this paper we cast the problem to a *reidentification* task, similarly to what has been done for person reidentification (Liu et al., 2017; Munjal et al., 2019; Xiao et al., 2017). Whereas Li et al. (2018) use the color and global features to make descriptors more robust, we propose to use them to filter the possible pairs to compare. Indeed, comparing every possible damage (which we consider as a *query*) with an insurance company's database containing all the previously checked claims could require an unsustainable number of comparisons. For this, we propose to filter the images based on the color, the brand, and the panel on which the damage is located, and we use this information to retrieve possible matches containing near duplicates to the new one. Zheng, Liang, et al. (2015) use a similar approach for the car-reidentification task. In their work, the car attributes are divided into two categories: special attributes and common attributes. Special

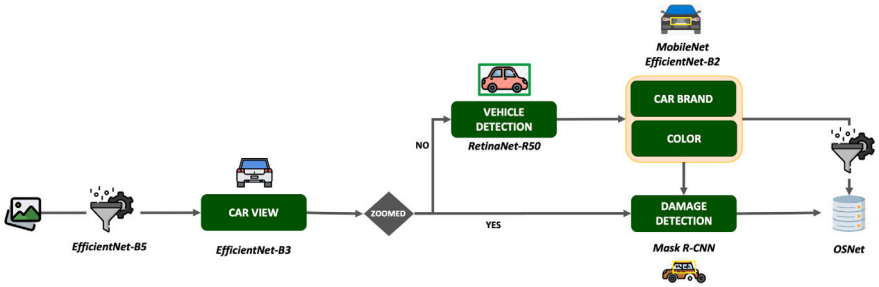


Fig. 2. Our proposed pipeline. Images are first filtered to select images that capture the exterior of the vehicle through an EfficientNet-B5 and then, based on the view of the vehicle, that is, the sides of the car visible in the image., with an EfficientNet-B3. Zoomed images that only capture details of the damage do not allow to extract vehicle information because the zoom on the damage makes it difficult, if not impossible, to extract information about the car or the location of the damage on the bodywork; therefore they are immediately sent to the damage detector module. If the image depicts the entire vehicle, we detect the car with a RetinaNet-R50, we extract the brand and color of the vehicle with an EfficientNet-B2 and MobileNet, respectively, and we locate the damage over the car bodywork with a Mask R-CNN. Images are then filtered based on these information and are finally sent to the damage reidentification module (i.e., OSNet).

attributes reflect the car's unique characteristics, such as individual paints or car damage, whereas common attributes denote the car's inherent appearance. Using specific attributes to re-rank results has been shown to increase retrieval performance. In our setting, however, we are interested in reidentifying damages, which represent one of the unique attributes of a car. Therefore, we chose to filter images based on their common characteristics to reidentify damage by only looking for it on vehicles of the same model, panel, and color as the query image.

Searching for damages is extremely challenging, as those may appear with a cluttered background and occlusion. In addition, the queried damage can appear in the gallery from different viewpoints, scales and lighting or reflection conditions, which makes this scenario very similar to that of the reidentification of objects, where an object can appear under different views and conditions. Most existing deep reidentification convolutional neural networks (CNNs) (Ahmed et al., 2015; Guo & Cheung, 2018; Li et al., 2014; Shen et al., 2018; Subramaniam et al., 2016; Varior et al., 2016; Wang et al., 2018) borrow architectures designed for generic object classification problems. However, these architectures are designed to treat objects with the exact and fixed shape that characterizes them. This does not apply to damages, which are harder to be matched to a shape. Each damage has unique distinguishing characteristics because of its typical irregular shape. Consequently, using CNNs designed to recognize objects of regular shape more easily leads the models to overfit on the training set without being able to really learn useful information for our task. An interesting idea to cope with this kind of problems comes from Zhou et al. (2019), who propose the OSNet, a network that learns multiscale features explicitly at each layer of the network. This is accomplished by a residual block composed of multiple convolutional streams, each detecting features at a certain scale. Then, a unified aggregation gate fuses multi-scale features with input-dependent channel-wise weights. To efficiently learn spatial-channel correlations and avoid overfitting, the building block uses pointwise and depthwise convolutions. Thanks to this structure, the OSNet turns out to be extremely lightweight and less prone to overfitting. For these reasons, in this paper, we propose a damage reidentifier based on an OSNet backbone and compare its performance with other state-of-the-art methods, specifically, Bursztein et al. (2021) and Deng et al. (2018).

In parallel with the drafting of this work, new reidentification strategies based on attention mechanisms have been proposed. He et al. (2021) introduced a transformer-based object reidentification framework, which is made of a patch module that rearranges patch embeddings by shift and shuffles operations. This results in robust features with increased discriminating ability and side information embeddings that counteract feature bias towards camera-view fluctuations by including these non-visual cues into learnable embeddings. Zhu et al. (2022) propose a similar approach, which uses a dual cross-attention learning algorithm to coordinate with self-attention learning, and they show that it reduces misleading attentions and diffuses the attention response to discover more complementary parts for recognition. These works are based on very deep models, which are helpful in tasks where many training examples are available. In our case, however, we do not have enough labeled data to justify the use of these models. Therefore, we decided to propose a less complex network, such as OSNet, which helps us control model overfitting.

Finally, in this work, as well as in the ones just discussed so far, we have a labeled dataset available and we treat the problem with a supervised approach. In addition, there are unsupervised approaches based on contrastive learning (Dai et al., 2022; Lin et al., 2020) or non-contrastive (Han et al., 2022) learning techniques. These approaches have not yet been explored on the damage reidentification task, and we leave this possibility open as a possible extension of this work in the future.

3. Approach

The images sent to insurance can be very different from each other. Some of these represent documents, other details of the bodywork or mechanical parts, and others could portray images of the interior or exterior of the vehicle. In addition to this, the data collected in different countries may have biases that differentiate them from those of others. To manage this complexity in a real system, we introduce an end-to-end pipeline for recognizing similar damages in a large gallery of collected images. Our system, illustrated in Fig. 2, is based on five main steps. Initially, images entering the pipeline are filtered to select only images of the exterior of the car. Then, the car view classification module classifies the sides of the car visible in the image, that is, front, back, front-left, back-right, and so on. Based on this classification, the zoomed images are directly sent to the damage recognition module, whereas the other images are used to extract the branding and color of the car. This information is useful for filtering the matches to be verified within the image database. The final module of the pipeline selects images of vehicles with damage similar to that of the newly uploaded images. Damages recovered from the system that exceed a certain similarity threshold are selected as potential copies of the damage and are then reported to the claim experts.

In the remaining of this section, we describe these components. We begin from the vehicle information extraction, which is used to reduce the number of damages to compare. Then, we discuss the damage detection and localization module, and, finally, we introduce our damage reidentification system.

3.1. Damage localization and claim-level feature aggregation

We integrate our damage reidentification system into a pipeline that deals with the identification of damages and the extraction of basic information on the claim under analysis. The first part of the pipeline

deals with filtering the images of the claims, selecting only those that portray the vehicle from the outside, as the damages that we analyze in this paper are damages that can only be found on the vehicle bodywork. To do this we use an EfficientNet-B5 (Tan & Le, 2020) trained on 11 classes (Documents, Odometer, Exterior, Interior, Mechanical Parts, Disassembled Parts, Display, three VIN classes, and Others) to select only images of the *exterior* (i.e., the Exterior class) of the vehicle. Next, we classify the vehicle view with an EfficientNet-B3 (Tan & Le, 2020) trained on 9 classes (Back, Back-Left, Back-Right, Left, Right, Front, Front-Left, Front-Right, and Zoomed). These two steps allow us to select only images of the exterior of the vehicle and distinguish different views of the bodywork of the car. These two steps are then followed by the extraction of the vehicle information and the recognition of the damage.

Vehicle information. With millions of new claims open every year, insurance companies collect a significant number of images. All this translates into having to compare each new damage with millions of other damages present in the database. In one year, this would mean making millions or billions of comparisons, which would be computationally prohibitively expensive. To reduce the complexity of these comparisons we propose a pipeline that extracts various information about the vehicle, which we use to reduce the search space: Although it is possible to perform fraud by reassembling a panel of one car on another one, or to claim damage on the same vehicle, these require that the target vehicle has the same *model* and *color* with the original one; thus the research can be limited to vehicles of the same brand, model, and color. of the same *model* and *color*. To automate this process, we propose the pipeline in Fig. 2. After the images have been filtered and classified according to the view of the vehicle, a RetinaNet-R50 (Lin et al., 2018) extracts the bounding boxes of the vehicle and the car's brand logo. These two pieces of information are then used to classify the brand and color of the car. For these two steps, we employ an EfficientNet-B2 (Tan & Le, 2020) for brand classification and a MobileNet (Howard et al., 2017) for color classification. The aforementioned models are all trained with *cross entropy* loss except for RetinaNet that is trained with the *focal loss* function introduced in Lin et al. (2018).

Damage detection and localization. Pictures of a claim must typically include both close-up shots of the damage and images with a distant view that contains the entire body of the car. These different perspectives allow on one hand the identification of the insured car and on the other hand, precise localization of the damage that has been reported. Identifying damage in an image is a key part of our pipeline as the entire reidentification system needs accurate damage identification to find any fraud attempts. However, it is important to note that this task is more challenging than traditional object detection problems, as the damage can have very different characteristics. In this work, we focus on the three most common damages: *scratches*, *dents*, and *cracks*. We treat this problem as a segmentation problem, in which we want to reconstruct the segmentation mask of the damage and its corresponding class. Mask R-CNN has proved extremely robust and accurate for this type of applications; therefore, we adopted it in our system with a ResNext-101 (Xie et al., 2017) *backbone* and we train this model to detect damages. Formally, we optimize the model parameters on each sampled *region of interest* with respect to the multitask loss introduced in He et al. (2018).

$$\mathcal{L} = L_{cls} + L_{bbox} + L_{mask}$$

The classification loss L_{cls} and the bounding box regression loss L_{bbox} are the same from the Faster R-CNN (Ren et al., 2016) architecture, whereas the mask component is the average binary cross-entropy loss used in the standard Mask R-CNN implementation (He et al., 2018).

Accurate damage detection is not enough in a production environment. Once the damage of a vehicle has been identified, claim experts usually need to verify that the damaged area corresponds to the one reported. Having this information not only allows us to have a more precise location of the damage but also allows us to exclude false matches with damages located in other positions of the car bodywork. Indeed, the same damage, even if disguised or reassembled on a new vehicle, will always be found on the same panel (component), which allows to exclude many other possible pairs. Thanks to the vehicle-view classification module, we can identify the location of the damage on one of the sides of the vehicle and thus reduce the possible matches to be identified in the database.

Claim-level aggregation. Although it is possible to train very robust deep-learning models, these will still be subject to some, albeit small, error rate. However, an error in the first part of the pipeline risks affecting the subsequent damage reidentification module. To reduce such errors, after an insured opens a claim for compensation and sends the images of the vehicle, we perform a *claim-level* refinement: Given two or more images belonging to the same claim, we select the brand that is predicted with higher confidence by the brand model across all images. Formally, given a set of brand predictions $B = \{f_b(x_1), \dots, f_b(x_m)\}$ for colored images $x_i \in \mathbb{R}^{H \times W \times 3}$, we take:

$$\arg\max_f \left| \{ f_b(x_i) \in B \mid x_i \in claim \} \right| \tag{1}$$

for any $m \geq 2$, where m represents the cardinality of the claim, and $f_b(x_i)$ represents the output confidence of the brand model for image i. The argmax operation selects the brand model that obtained the higher confidence across all the m images of the claim.

Furthermore, we apply the claim-level logic for the classification of the color of the car as well. At the image level, we apply a threshold τ to select the color predicted by the model. Below this confidence, we also consider the second class with the highest score, that is:

$$f_c(x_i) = \begin{cases} C_k, & \text{if } C_k \geq \tau. \\ (C_k, C_{k-1}), & \text{otherwise.} \end{cases} \tag{2}$$

for any ordered prediction $\{C_1, \dots, C_k\}$ where C_k is the highest value from the softmax function. Finally, given a set of color predictions $C = \{f_c(x_1), \dots, f_c(x_m)\}$ on the images of the claim, the most voted color is chosen as the color of the car. Formally:

$$\arg\max_f \left| \{ f_c(x_i) \in C \mid x_i \in claim \} \right| . \tag{3}$$

This phase of aggregation of the claims allows for the extraction of color and model information directly from the images, thus verifying that the information on the reported vehicle is consistent with the data of the insured vehicle and, finally, allows, as explained, to reduce the number of necessary comparisons within the database.

3.2. Damage reidentification

Damages identified by the damage localization and vehicle information modules are ready to be reidentified via the damage-similarity module. Different shots of the same damage can be very different. The same damage can be captured in different lighting conditions, reflections, zoom levels, partial obstructions, and perspectives. The damage reidentification module must therefore be robust to all these variables. We chose to build our reidentification module on top of the OSNet of Zhou et al. Zhou and Xiang (2019), Zhou et al. (2019, 2021), which proved to be very effective for people's reidentification task. In our setup, the OSNet works as a feature extractor that maps input damage into an *embedding* space. Hence, the damages are directly compared in this space through *cosine similarity*. Unlike Li et al. (2018) who propose to add global image features that represent the vehicle's color and view, in our case we use the car view, color, and brand information to reduce the number of required comparisons. This allows us on the one hand to significantly reduce the time required to reidentify the damage and on the other hand to eliminate some biases such as the vehicle's view or color from the vector representation of the damage. Considering

the heterogeneity of the photos of the same damage, many photos may contain information that is not useful for the reidentification of the damage. Therefore, we have chosen to crop the image around the damage to only include this information when comparing two damages. This allows us to eliminate unwanted noise and focus solely on the areas of interest.

The goal of our damage reidentification module is to learn an embedding function $f_\theta(x) : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^E$ that maps semantically similar points from the data manifold in $\mathbb{R}^{H\times W\times 3}$ to close points in $\mathbb{R}^E$ and different points in $\mathbb{R}^{H\times W\times 3}$ to distant points in $\mathbb{R}^E$. Formally, let $D(f_\theta(x_i), f_\theta(x_j)) : \mathbb{R}^E \times \mathbb{R}^E \rightarrow \mathbb{R}$ be a metric function measuring distances in the embedding space. We train our model to minimize the *hard triplet loss* (Hermans et al., 2017) function, which for each sample in a batch, selects the hardest positive and the hardest negative samples *within the batch*. We create the batches by randomly sampling D different damages, and again randomly sampling K images of the same damage, thus obtaining a batch of DK images. Then, we train the model to minimize the following loss function for each batch X:

$$\mathcal{L}_B(\theta, X) = \sum_{i=1}^{D}\sum_{a=1}^{K}\Bigg[m + \overbrace{\max_{p=1\dots K} D(f_\theta(x_a^i), f_\theta(x_p^i))}^{\text{hardest positive}} - \underbrace{\min_{\substack{j=1\dots D\\ n=1\dots K\\ i\neq j}} D(f_\theta(x_a^i), f_\theta(x_n^j))}_{\text{hardest negative}} \Bigg]_+, \tag{4}$$

where a data point x_j^i corresponds to the ith image of the jth damage in the batch. This loss ensures that, given an *anchor* point x_a, the projection of a positive point x_p representing the same damage j is closer to the anchor's projection than that of a negative point x_n belonging to another class d, by at least a margin m. The margin guarantees that in the end, points that are sufficiently close to each other will end up belonging into the same cluster, representing multiple copies of the same damage.

We choose the cosine similarity as our distance metric:

$$D(f_\theta(x^i), f_\theta(x^j)) = \frac{f_\theta(x^i)\cdot f_\theta(x^j)}{\|f_\theta(x^i)\|\|f_\theta(x^j)\|}$$

and use the following similarity score:

$$S(f_\theta(x^i), f_\theta(x^j)) = \begin{cases} 0, & \text{if } D(f_\theta(x^i), f_\theta(x^j)) < \zeta, \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

Then two damages x^i and x^j are considered the same damage if $S(f_\theta(x^i), f_\theta(x^j)) = 0$.

4. Implementation details

All the experiments that we present in Section 5 were conducted on an Azure Standard NCasT4-v3 series virtual machine with a 16 GB NVIDIA T4. Next we provide some implementation details on the models of Section 3. Then, in Section 4.1 we present the datasets for training and evaluating our approach and in Section 4.2 our evaluation metrics.

We trained the Filter, the car-view and the vehicle-detection models on 684×684 images (introduced in Section 4.1) with the Adam (Kingma & Ba, 2014) optimizer and a learning rate set to 0.0001. We trained the models to minimize the cross-entropy loss on batches of 4 images. Instead, we trained the brand and the color classification models with the same configuration but on batches of 32 images of size 224×224 pixels. We trained the damage modules on batches of 4 images with basic Detectron2 (Wu et al., 2019) configuration. Finally, we trained the damage similarity module on 256×256 images with a larger batch of 64 images and basic learning rate set to 0.0003. We projected the images onto a 512–dimensional embedding space and, after performing experiments, we chose $m = 0.3$, $\tau = 0.5$, and $\zeta = 0.5$ for Eqs. (1), (2) and (5) respectively.

4.1. Datasets

The training and testing of all pipeline components that we have introduced so far require to tackle several tasks. In industrial applications, data preparation requires a major effort to be able to structure datasets useful for model training. For this reason, we have decided to dedicate this section of the paper to the description of the datasets built to train the various components of the pipeline introduced so far. Unfortunately, we cannot release all the data used for each component as they are subject to privacy and covered by trade secrets, but we will describe them in detail and report their characteristics summarized in Fig. 3. In addition, to facilitate reproducibility of results, we also introduce a new set of tests for the damage reidentification task. The test set and the proposed models will be released publicly. For all datasets, unless otherwise specified, we use 90% of the data for training and the remaining 10% for testing. Finally, because we propose a supervised method for damage reidentification, we use the COCO Annotator (Brooks, 2019) tool to annotate the damages and components of the car in the images.

Damage detection and localization. To identify and locate the damage to the vehicle we have introduced two components. The first deals with classifying the vehicle view and the second identifies the damage. For the former, we built a dataset of 14,054 images. Each of these has been annotated to classify the following 9 views: back, back-left, back-right, left, right, front, front-left, front-right, and zoomed. Fig. 3(b) shows the distribution of each element. For the second task, we created a dataset of 3818 examples of scratches, cracks, and dents. As shown in Fig. 3(f), scratches are numerically much more frequent. Consequently, to balance the number of samples across all the classes, in the training phase we increase the crack and dent by applying a data augmentation strategy by introducing random flipping, rotation, saturation, contrast and brightness. Because the damage can only be present in photos of vehicle exteriors, in Fig. 3(a) we report the distribution of the dataset used to train the filter. The dataset contains many other classes than External, which are used for other purposes that are beyond the scope of this work.

Vehicle information. The detection and extraction of the vehicle and its basic information such as the brand and color is another fundamental step of the reidentification system. The first stage, therefore, requires vehicle detection. To this end, the model was trained on a dataset of 2036 images. Instead, the color detection model is trained on a dataset of 210 images containing 13 colors. Finally, a dataset of 36 brands with a total of 37,000 images was created for the brand identification task. For this last dataset, we use the 80% of the data for training and the remaining 20% for testing. Figs. 3(e), 3(d) and 3(c) show the statistics of these datasets.

Damage reidentification. To train our damage reidentification model we built a dataset, shown in Fig. 3(g), of 57,950 images. Of these, we use 90% for training and the remaining 10% for validation. This dataset contains 11,571 possible matches and is constructed from several sources that include images from real claims (marked as *From countries* in Fig. 3(g)), two internal datasets of damages (labeled as Source 1 and 2), and a synthetic dataset that was constructed as follows: First, we manually extract 74 real damages from images of damaged vehicles using the GIMP software (The GIMP Development Team, 2019). The damages were extracted with a transparent background to remove details of the original bodywork. Then, for each damage, we automatically paste it on a car identified through our vehicle detection system. For each vehicle image, we create 7 different versions of the damage by applying it in 5 different positions and creating a perspective change and an affine transformation of the damage.

We evaluate the model on two test sets. The first contains 385 images with a total of 139 unique damages with at least 2 matches per

(a) Filter dataset.

(b) Car-view dataset.

(c) Brand-recognition dataset.

(d) Color-detection dataset.

(e) Vehicle-detection dataset.

(f) Damage-detection dataset.

(g) Damage-reidentification dataset.

Fig. 3. Distribution of the training and evaluation datasets used for the components of the pipeline. For each dataset, we show the classes and corresponding number of samples per class. For the damage reidentification dataset (Fig. 3(g)) we report several sources used to construct the dataset, and the minimum (Min) and the maximum (Max) number of matches for each image.

damage and up to 7 matches. The second,[1] contains 567 images with 420 possible matches with at least 38 matches for each damage. The public test set was collected by acquiring images of 42 vehicles with five different smartphones. The images contain zoomed and nonzoomed views of the vehicle and were captured in different lighting, dirt, and reflection conditions to simulate the images sent by policyholders as realistically as possible.

4.2. Evaluation metrics for damage reidentification

We conclude this section with a brief discussion on the evaluation metrics adopted to evaluate the reidentification system. The task is a reidentification problem, and as such, we use the most commonly adopted metrics to evaluate these tasks (Ye et al., 2020). In addition, we add two metrics that we used to evaluate the system and which proved useful for scaling the system into production. We call them *recall-at-top-k* ($recall_k$) and *F1-at-top-k* ($F1_k$).

The first reference metric is the *cumulative matching characteristics* (CMC). CMC-k (also known as Rank-k matching accuracy Wang et al., 2007) denotes the likelihood that a correct match will appear in the top-k ranked retrieved results. Because it only examines the first match in the assessment process, CMC is accurate when only one ground truth exists for each query. However, in a real setting, the image database typically comprises many ground truths, so CMC cannot completely reflect a model's discriminability across numerous matches. Therefore, we use the Mean Average Precision (mAP, Zheng, Shen, et al., 2015). It assesses average retrieval performance with numerous ground facts. Originally, it was frequently used in image retrieval. It can address the issue of two systems doing equally well in searching for the first ground truth but having varied retrieval abilities for additional challenging matches in reidentification evaluation.

Finally, we use $recall_k$. Because we are interested in finding the highest number of matches, the recall allows us to measure the number

[1] URL will appear here after publication.

Table 1
Damage detection. model trained with different learning rate (LR) strategies.

LR strategy	IoU	mAP
Step LR	10.0%	64.7%
	25.0%	45.0%
Cosine LR	10.0%	67.6%
	25.0%	51.9%
Aug. Cosine LR	10.0%	**69.6%**
	25.0%	51.9%

of matches correctly identified against the total number of possible matches. However, wanting to calculate this value with respect to the top-k, we apply a change to the recall.

$$recall_k = \frac{\text{TP in the top-}k\text{ results for each query}}{\text{Max TP}}$$

where

$$\text{Max TP} = Q - \sum_{q=1}^{Q} max\{TP_q - k, 0\}$$

with TP_q representing the true positives (TP) of the actual query q and Q representing the number of queries. From this definition, we can finally define the $F1_k$ as follows.

$$F1_k = 2\frac{precision \cdot recall_k}{precision + recall_k} \tag{6}$$

In Section 5.3 we evaluate our damage similarity system on all these metrics and show how they help to scale our solution to a production setting.

5. Results

In this section, we evaluate the performance of the proposed pipeline. Before discussing the performance of the reidentification model, shown in Table 2, we begin by reporting the performance of the vehicle information, and damage detection and localization components that contribute to the functioning of the damage reidentification system. Finally, we conclude by reporting the performance of the damage reidentification model and presenting the challenges faced in scaling the model in a production pipeline.

5.1. Vehicle detection and localization

Extracting basic vehicle information allows us to reidentify damage more accurately. Although it is interesting to reduce the error rate of the reidentification system as much as possible, *an error rate of 1% on 2 million images still implies a very high number of false alarms*. In addition, it leads to a very high number of images to compare. As explained, however, it is possible to reduce the complexity simply by removing all possible unnecessary pairs by extracting vehicle information. First, the vehicle identification pattern introduced in Section 3.1 achieves 85.0% accuracy. Once the vehicle has been localized, we identify the color and brand of the car. The brand recognition model accurately classifies 98.0% of the images analyzed, and the color model achieves 87.0% accuracy.

In Section 6 we show the benefits of these components by scaling into production.

5.2. Damage detection

The damage detector is a key component of our pipeline, as damage reidentification would not be effective without accurate damage recognition. As explained previously, in this work we focus on *cracks*, *dents*, and *scratches*. However, we are not interested in the classification of damage, that is, the correct identification and classification of the damage as a crack, dent, or scratch, but we limit the analysis to recognizing each of these classes as damage. In spite of this, we still consider the classification task to be very interesting and at the same time complex; we, therefore, leave the possibility to investigate this problem in the future.

Unlike many other object detection tasks, damage can be very heterogeneous, with different shapes, colors and sizes. Especially in some conditions of light and reflection on the bodywork (see Fig. 1), recognizing some types of damage (especially the dents) can be very complex even for the most experienced claim experts. For this, proper tuning of the model hyperparameters can help to significantly improve performance. In our experiments, we compared three different update configurations of learning-rate. Table 1 shows the comparison in terms of IoU and mAP between three learning rate strategies: (1) *Step LR*, the learning speed of each group of gamma parameters decays at each step size epoch, (2) *Cosine LR*, setting the learning speed of each parameter group using a cosine annealing program, and (3) *Aug. Cosine LR*, the Cosine LR scheduler with data augmentation. From these experiments, the Cosine LR obtains better performances than the Step LR, and the data augmentation further contributes to improving the overall robustness of the model.

Ultimately, recognizing the damage is not sufficient to understand where it is located on the vehicle body. For this, we also report the performance of the filter and car view models. The first means that the damage detection model receives in input only images of the exterior of the vehicle which can therefore contain damages. This model achieves 96.0% of accuracy. The car view model allows to identify the location of the damage on the vehicle body and to reduce the comparisons necessary to identify possible matches. In this case the model achieves 90.0% of accuracy.

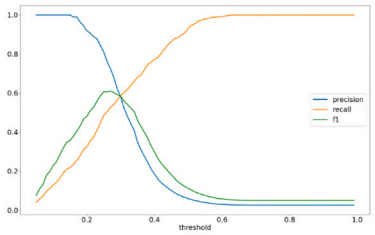
5.3. Damage reidentification

The inspection of the damage similarity is the final step in our pipeline. Damages and information extracted from previous modules can be used to identify possible fraud attempts. In this section, we present the experiments that we conducted on the model and in Section 6 we show the performance of the model when aggregated with information extracted from the other components of the pipeline. Fig. 4 shows some reidentification examples of our system.

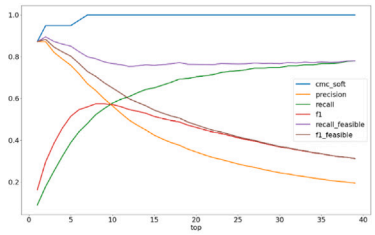
To evaluate the performance of the proposed solution, we compare it with two models commonly used for image similarity tasks: Tensorflow Similarity (Bursztein et al., 2021) and Arcface (Deng et al., 2018). For all models, we report the results on the public and private test sets introduced in Section 4.1. As shown in Table 2a and b, our similarity model achieves the best performance across both datasets. Arcface achieves higher recall on the public test set, but our proposed model still outperforms all the others in terms of mAP, CMC and top-k $F1_k$. The F1 score is the most important metric for us to take into account. If the recall indicates the number of correctly reidentified damages, to apply the system in a real scenario, we must be sure that the ratio between these and the number of false positives is not too high; otherwise, we would provide a system that correctly identifies an increased number of matches together with an excessive number of false alarms, making our system inapplicable in practice.

Fig. 5(a) reports the precision–recall curve obtained at different threshold values of the model's predicted class scores. This plot is essential to deploy such a model into production, because it allows to measure the tradeoff between these two metrics. In a real setting, the balance between these two metrics is very important. A system with higher precision is preferred over one with higher recall. In fact, it is very important to have a small number of false positives with respect to the total number of alerts: Because each of the alerts is verified by a claim expert, a high number of false positives would require the verification of too many alarms, thus raising the costs of maintaining

Fig. 4. Sample queries and matching images in the public test set. The leftmost images are query images, followed by the highest similarity matches. Images framed in green are correct reidentifications, whereas those in purple indicate errors. The similarity score is also shown above each image. Even in cases of error, the model identifies images that are very similar to the query one.



(a) Precision–recall curve and F1 at different thresholds.



(b) The different evaluated metrics at top-k.

Fig. 5. Evaluation metrics of the damage reidentification model on the public test set. Fig. 5(a) shows the precision–recall curve and F1 score obtained at different thresholds. Fig. 5(b) shows the variation of the performances with respect to the top-k predictions for different values of k.

Table 2
Evaluation of the damage-similarity model on the public test and private test sets.

Model	mAP	CMC	Top 5 $F1_k$	Top 10 $F1_k$	$recall_k$ 95	$recall_k$ 90	$recall_k$ 50
TF similarity (Bursztein et al., 2021)	50.8%	76.9%	56.9	45.2%	12.1%	19.7%	43.6%
Arcface (Deng et al., 2018)	63.7%	84.6%	72.1	58.2%	**33.1%**	**36.7%**	**60.6%**
Ours	**71.5%**	**87.2%**	**80.2**	**65.0%**	29.1%	35.4%	**60.6%**

(a) Public test set.

Model	mAP	CMC	Top 5 $F1_k$	Top 10 $F1_k$	$recall_k$ 95	$recall_k$ 90	$recall_k$ 50
TF similarity (Bursztein et al., 2021)	61.4%	61.4%	33.9	22.5%	–	15.4%	39.1%
Arcface (Deng et al., 2018)	64.4%	64.4%	35.6	24.2%	32.0%	35.2%	49.0%
Ours	**79.2%**	**79.5%**	**43.8**	**28.1%**	**45.5%**	**49.8%**	**72.3%**

(b) Private test set.

the process. However, retrieving all potential fraud attempts is also very important.

Fig. 5(b) adds another important ingredient to scale into production. The system maintains a very high $recall_k$ within the top-5 and beyond. This is an encouraging result, as it suggest that within five possible similarity alerts there will be a very high probability of encountering a correct match.

There is another key element to consider in evaluating the applicability of such a model in an industrial setting. The end users will be anti-fraud experts, but as such, ignore how to interpret deep-learning models. What is crucial is that users do not perceive alarms as completely random. Interpretability is very important. As mentioned, a limited number of errors is acceptable, but for the system to be really used it is necessary that even the errors are somehow interpretable. Fig. 4 shows some examples of matches produced by our system. Green framed images indicate correct matches and purple frames indicate errors. Although some recovered images are incorrect, the errors are still acceptable as they include images that are very similar to query images.

Fig. 6 shows an embedding obtained through a projection of the features through UMAP (McInnes et al., 2018). Dots of the same color represent the real matches. Interestingly, the model learns to correctly map similar damage that is very close to each other. What is further interesting is that many false alarms consist of examples that are visually very similar to the input one. In fact, the model maps nearby images of cars of the same model or very similar models and of the same color. Even though damage is therefore a key component of learning, it is not the only feature used by the model. Added to this, Fig. 6 shows the attention maps of three images. The activations are mostly concentrated around the damage, which confirms that the model is correctly looking at the damaged area of the picture to make a decision. The analysis of the attention maps suggests that there are some problems that still need to be solved. In many cases, the activations are stronger around the vehicle's escape lines. These are in fact very similar to damage, especially with respect to scratches. We leave the solution of this issue for future development.

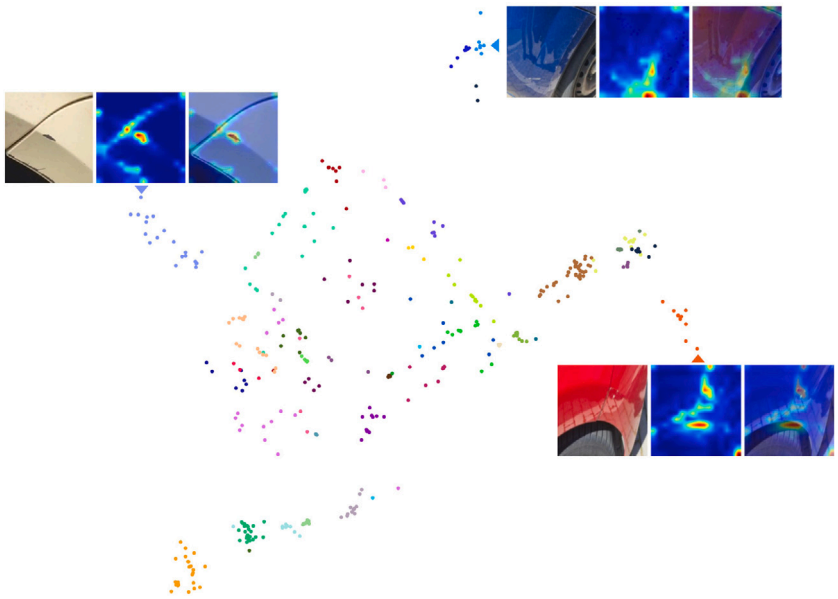


Fig. 6. Embedding projection in a two-dimensional space using UMAP (McInnes et al., 2018). Points of the same color represent the same damages. We show three sample damages with their corresponding heatmap. The third column images show the heatmaps overlapped over the images for a better visualization of the activations. The model activations correctly focus on the damaged parts of the vehicle. Added to this, images of the same damages are correctly mapped one to each other in the embedding space.

6. Discussion

The proposed solution allows for the identification of possible duplicate damages with acceptable performance in the test phase. However, as mentioned, it is important to be able to apply these solutions in a real scenario. The largest insurance companies operate in several countries around the world. This means being exposed to a huge number of possible variations in the image acquisition processes as well as in the characteristics of the insured vehicles. This implies that the performance of the proposed solution may vary depending on the countries where it is applied. In general, it is possible to identify the two most frequent causes of the variation in performance: (1) the *image quality*, which is higher in some countries and very low in others, and (2) the *average number of images* acquired for each claim, which can vary depending on the regulations applied in the various countries. Therefore, the main idea behind this work, is to support the damage reidentification module with the vehicle information extracted by the other components of the pipeline. Fig. 7 show the average effect of filtering across several countries. Despite the good performances in an experimental setting, the proposed damage reidentification system would still produce an average 90% of *false positives* (FP). Obviously this is not acceptable. To reduce the alarms further, it is necessary to integrate the information extracted from the images with the Vehicle Identification Number (VIN) data. This is an identification number that acts as the car's fingerprint, as there are no two vehicles on the road with the same VIN. A VIN consists of 17 characters (digits and uppercase letters) which serve as a unique identifier for the vehicle. A VIN shows the car's unique features, specifications, and manufacturer. The VIN can be used to track recalls, registrations, warranty claims, theft, and insurance coverage. By integrating our proposed method with the car model filtering extracted through the VIN leads to 65% of false alarms. By further refining the filter by restricting the search to the single-vehicle, the FP is further reduced by up to 18%, which represents a 72% reduction of possible alerts.

We hope that this analysis will stimulate the interest of the scientific community in this type of problems. The results show that despite good performance in the experimental settings, it is still difficult to use a system based solely on image analysis.

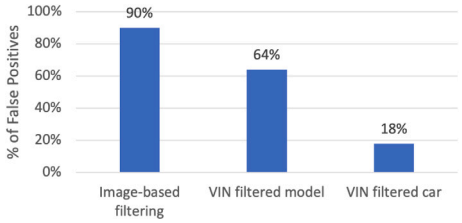


Fig. 7. Average percentage of false positives across two European countries. Despite the good performance in an experimental setting, the proposed system still produces an average 90% of FPs. To apply the system into production, we still need data external to the images, for instance, the VIN.

7. Conclusions

In this paper, we proposed an automatic system for identifying fraud attempts in the insurance sector. The system allows, once the images of an accident have been received, to extract information about the vehicle and to identify similar damages within a collection of images. The solution has been validated by comparing the proposed method with two state-of-the-art solutions for image similarity estimation. We showed that our method outperforms current solutions and we validated the solution both quantitatively and qualitatively. The visualization of the embedding and the attention maps, confirm that the system is able to project input images into a space in which similar damages are mapped closer. In addition to this, we have released a new set of benchmark tests, which we hope will fuel the debate in this new field of application of image retrieval. Finally, we discussed the challenges that need to be addressed to scale these systems into a production environment. The recognition of damages and the aggregation of information extracted from the vehicle allow a significant reduction of comparisons in the database, however, the number of false alarms remains too high, showing that more research is needed to solve this hard problem.

As highlighted in our experiments, the proposed solution can still be improved. The current solution is heavily dependent on the data it is trained on. Having approached the problem in a supervised way, the data used in training are rather limited. Future work can, therefore,

focus on the design of self-supervised learning techniques that allow training on a much greater number of images. As shown in our experiments, adding data augmentation improves performance by a good margin. Thus, future work can analyze this setting in more detail to train the unlabeled system more effectively.

Data and code availability

The code of all models and the dataset is publicly available on our Github repository.[2]

CRediT authorship contribution statement

Luca Maiano: Conceptualization, Methodology, Data curation, Software, Investigation, Validation, Formal analysis, Writing, Visualization, Writing – original draft, Writing – review & editing. **Antonio Montuschi:** Project administration, Conceptualization, Methodology, Resources, Funding acquisition. **Marta Caserio:** Project administration, Conceptualization, Methodology, Resources, Funding acquisition. **Egon Ferri:** Conceptualization, Methodology, Data curation, Software, Investigation, Validation. **Federico Kieffer:** Conceptualization, Methodology, Data curation, Software, Investigation, Validation. **Chiara Germanò:** Conceptualization, Methodology, Data curation, Software, Investigation, Validation. **Lorenzo Baiocco:** Conceptualization, Methodology, Data curation, Software, Investigation, Validation. **Lorenzo Ricciardi Celsi:** Project administration, Funding acquisition. **Irene Amerini:** Supervision. **Aris Anagnostopoulos:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Antonio Montuschi reports a relationship with Assicurazioni Generali that includes: employment. Marta Caserio reports a relationship with Assicurazioni Generali that includes: employment. Lorenzo Ricciardi Celsi reports a relationship with ELIS Innovation Hub that includes: employment. Federico Kieffer reports a relationship with ELIS Innovation Hub that includes: employment. Egon Ferri reports a relationship with ELIS Innovation Hub that includes: employment. Lorenzo Baiocco reports a relationship with ELIS Innovation Hub that includes: employment. Chiara Germano reports a relationship with ELIS Innovation Hub that includes:.

Acknowledgments

This research was supported by the Assicurazioni Generali, the Elis Innovation Hub, the ERC Advanced Grant 788893 AMDROMA, the EC H2020RIA project "SoBigData++" (871042), the PNRR MUR project PE0000013-FAIR," the PNRR MUR project IR0000013-SoBigData.it, and the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union – NextGenerationEU.

References

Ahmed, E., Jones, M., & Marks, T. K. (2015). An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Association of British Insurers (ABI) (2022). 8 myths about insurance fraud busted. https://www.abi.org.uk/products-and-issues/topics-and-issues/fraud/8-myths-about-insurance-fraud/. (Accessed 10 March 2022).

Bandi, H., Joshi, S., Bhagat, S., & Deshpande, A. (2021). Assessing car damage with convolutional neural networks. In *2021 International conference on communication information and computing technology* (pp. 1–5). IEEE.

Brooks, J. (2019). COCO annotator. https://github.com/jsbroks/coco-annotator/.

Bursztein, E., Long, J., Lin, S., Vallis, O., & Chollet, F. (2021). TensorFlow similarity: A usable high-performance metric learning library. Fixme. URL: https://github.com/tensorflow/similarity.

Cantarini, G., Noceti, N., & Odone, F. (2020). Boosting car plate recognition systems performances with agile re-training. In *2020 IEEE 4th international conference on image processing, applications and systems* (pp. 102–107). IEEE.

Dai, Z., Wang, G., Yuan, W., Zhu, S., & Tan, P. (2022). Cluster contrast for unsupervised person re-identification. In *Proceedings of the Asian conference on computer vision* (pp. 1142–1160).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). http://dx.doi.org/10.1109/CVPR.2009.5206848.

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2018). ArcFace: Additive angular margin loss for deep face recognition. arXiv. http://dx.doi.org/10.48550/ARXIV.1801.07698. URL: https://arxiv.org/abs/1801.07698.

Djara, T., Assogba, M. K., & Vianou, A. (2017). An approach for benin automatic licence plate recognition. *International Journal of Image Processing (IJIP)*, *11*(2), 25.

Etomi, E. E., & Onyishi, D. U. (2021). Automated number plate recognition system. *Tropical Journal of Science and Technology*, *2*(1), 38–48.

FBI (2022). FBI report on insurance fraud. https://www.fbi.gov/stats-services/publications/insurance-fraud. (Accessed 10 March 2022).

Guo, Y., & Cheung, N.-M. (2018). Efficient and deep person re-identification using multi-level similarity. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Han, D., Liu, W., Zou, M., & Liu, B. (2022). Non-contrastive nearest neighbor identity-guided method for unsupervised object re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 1. http://dx.doi.org/10.1109/TCSVT.2022.3224994.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). Mask R-CNN. arXiv:1703.06870.

He, S., Luo, H., Wang, P., Wang, F., Li, H., & Jiang, W. (2021). TransReID: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 15013–15022).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv. http://dx.doi.org/10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

Hermans, A., Beyer, L., & Leibe, B. (2017). In defense of the triplet loss for person re-identification. arXiv. http://dx.doi.org/10.48550/ARXIV.1703.07737. URL: https://arxiv.org/abs/1703.07737.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.

Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2016). Spatial transformer networks. arXiv:1506.02025.

Jain, V., Sasindran, Z., Rajagopal, A., Biswas, S., Bharadwaj, H. S., & Ramakrishnan, K. (2016). Deep automatic license plate recognition system. In *Proceedings of the tenth Indian conference on computer vision, graphics and image processing* (pp. 1–8).

Khan, M. H.-M., Hussein Sk Heerah, M. Z., & Basgeeth, Z. (2021). Automated detection of multi-class vehicle exterior damages using deep learning. In *2021 International conference on electrical, computer, communications and mechatronics engineering* (pp. 01–06). http://dx.doi.org/10.1109/ICECCME52200.2021.9590927.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv. http://dx.doi.org/10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

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

Li, P., Shen, B., & Dong, W. (2018). An anti-fraud system for car insurance claim based on visual evidence. arXiv preprint arXiv:1804.11207.

Li, W., Zhao, R., Xiao, T., & Wang, X. (2014). DeepReID: Deep filter pairing neural network for person re-identification. In *2014 IEEE conference on computer vision and pattern recognition* (pp. 152–159).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection. arXiv:1708.02002.

Lin, Y., Xie, L., Wu, Y., Yan, C., & Tian, Q. (2020). Unsupervised person re-identification via softened similarity learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Liu, H., Feng, J., Jie, Z., Jayashree, K., Zhao, B., Qi, M., Jiang, J., & Yan, S. (2017). Neural person search machines. In *2017 IEEE international conference on computer vision* (pp. 493–501). Los Alamitos, CA, USA: IEEE Computer Society, http://dx.doi.org/10.1109/ICCV.2017.61, URL: https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.61.

Lubna, Mufti, N., & Shah, S. A. A. (2021). Automatic number plate recognition:A detailed survey of relevant algorithms. *Sensors*, *21*(9), http://dx.doi.org/10.3390/s21093028, URL: https://www.mdpi.com/1424-8220/21/9/3028.

Malik, H. S., Dwivedi, M., Omakar, S., Samal, S. R., Rathi, A., Monis, E. B., Khanna, B., & Tiwari, A. (2020). Deep learning based car damage classification and detection. EasyChair Preprint.

Manana, M., Tu, C., & Owolawi, P. A. (2021). Edge-based licence-plate template matching for identifying similar vehicles. *Vehicles*, *3*(4), 646–660. http://dx.doi.org/10.3390/vehicles3040039, URL: https://www.mdpi.com/2624-8921/3/4/39.

[2] https://github.com/Assicurazioni-Generali/damage-similarity

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv. http://dx.doi.org/10.48550/ARXIV.1802.03426. URL: https://arxiv.org/abs/1802.03426.

Munjal, B., Amin, S., Tombari, F., & Galasso, F. (2019). Query-guided end-to-end person search. In *The IEEE conference on computer vision and pattern recognition*.

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

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. arXiv:1506.02640.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv:1506.01497.

Shen, Y., Xiao, T., Li, H., Yi, S., & Wang, X. (2018). End-to-end deep kronecker-product matching for person re-identification. In *2016 IEEE conference on computer vision and pattern recognition*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv. http://dx.doi.org/10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556.

Subramaniam, A., Chatterjee, M., & Mittal, A. (2016). Deep neural networks with inexact matching for person re-identification. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems. Vol. 29*. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2016/file/e56b06c51e1049195d7b26d043c478a0-Paper.pdf.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. arXiv. http://dx.doi.org/10.48550/ARXIV.1512.00567. URL: https://arxiv.org/abs/1512.00567.

Tan, M., & Le, Q. V. (2020). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv:1905.11946.

The GIMP Development Team (2019). GIMP. URL: https://www.gimp.org.

Varior, R., Haloi, M., & Wang, G. (2016). Gated siamese convolutional neural network architecture for human re-identification. In *Proceedings of the 14th European conference on computer vision. Vol. 9912* (pp. 791–808). http://dx.doi.org/10.1007/978-3-319-46484-8_48.

Wang, Y., Chen, Z., Wu, F., & Wang, G. (2018). Person re-identification with cascaded pairwise convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, X., Doretto, G., Sebastian, T., Rittscher, J., & Tu, P. (2007). Shape and appearance context modeling. In *2007 IEEE 11th international conference on computer vision* (pp. 1–8). http://dx.doi.org/10.1109/ICCV.2007.4409019.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

Xiao, T., Li, S., Wang, B., Lin, L., & Wang, X. (2017). Joint detection and identification feature learning for person search. In *CVPR*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492–1500).

Yaacob, N. L., Alkahtani, A. A., Noman, F. M., Zuhdi, A. W. M., & Habeeb, D. (2021). License plate recognition for campus auto-gate system. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*(1), 128–136.

Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., & Hoi, S. C. H. (2020). Deep learning for person re-identification: A survey and outlook. arXiv. http://dx.doi.org/10.48550/ARXIV.2001.04193. URL: https://arxiv.org/abs/2001.04193.

Zhang, Q., Chang, X., & Bian, S. B. (2020). Vehicle-damage-detection segmentation algorithm based on improved mask RCNN. *IEEE Access*, *8*, 6997–7004. http://dx.doi.org/10.1109/ACCESS.2020.2964055.

Zheng, Q., Liang, C., Fang, W., Xiang, D., Zhao, X., Ren, C., & Chen, J. (2015). Car re-identification from large scale images using semantic attributes. In *2015 IEEE 17th international workshop on multimedia signal processing* (pp. 1–5). http://dx.doi.org/10.1109/MMSP.2015.7340861.

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., & Tian, Q. (2015). Scalable person re-identification: A benchmark. In *2015 IEEE international conference on computer vision* (pp. 1116–1124). http://dx.doi.org/10.1109/ICCV.2015.133.

Zhou, K., & Xiang, T. (2019). Torchreid: A library for deep learning person re-identification in pytorch. arXiv preprint arXiv:1910.10093.

Zhou, K., Yang, Y., Cavallaro, A., & Xiang, T. (2019). Omni-scale feature learning for person re-identification. In *ICCV*.

Zhou, K., Yang, Y., Cavallaro, A., & Xiang, T. (2021). Learning generalisable omni-scale representations for person re-identification. *TPAMI*.

Zhu, H., Ke, W., Li, D., Liu, J., Tian, L., & Shan, Y. (2022). Dual cross-attention learning for fine-grained visual categorization and object re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4692–4702).

Zhu, X., Liu, S., Zhang, P., & Duan, Y. (2019). A unified framework of intelligent vehicle damage assessment based on computer vision technology. In *2019 IEEE 2nd international conference on automation, electronics and electrical engineering* (pp. 124–128). IEEE.