Applied Sim-To-Real Transfer for Damage Estimation

Pascal Godejohann[1], Marc Mueller[2], Pascale Maul[2], Eva Pigova[2], Lefteris Stamatogiannakis[2], and Thomas Fischer

(1) HDI Deutschland AG, HDI-Platz 1, 30659 Hannover, Germany, e-mail: pascal.godejohann@hdi.de
(2) BeamNG GmbH, Domshof 17, 28195 Bremen, Germany, e-mail: {pmaul, epigova, mmueller, estama, tfischer}@beamng.gmbh

Abstract - *After an accident, filing insurance claims is a time consuming and often complicated process, requiring collaborative effort by the insured, the claims appraiser and the insurance company. Multiple approaches to (partially) automate the claims submission process have been explored. While annotated real-world data is available for insurance companies through the documentation of previous claims process, the EU privacy policy prevents companies to use them for the training of neural networks (Kop, 2020). This work explores the use of synthetic images for automated damage evaluation of real-world data. Thus, the authors present their two-fold contribution: $a)$ a data generation framework based on the BeamNG.tech simulator and $b)$ a deep learning computer vision system for damage estimation.*

Keywords: *deep learning, damage evaluation, sim-to-real transfer*

1. Introduction

1.1. Background

Traditional insurers often retain an analog and lengthy process for filing claims. Even more so in cases where the documentation provided by the insured party is not sufficient to evaluate the damage and in severe accidents. The insured needs to notify their insurer through a hotline, submit a formal registration of the accident and has to make an appointment with an independent claims adjuster. After the appraisal, the claims adjuster then in turn submits another report with a damage estimate and assessment to the insurer for approval. Only then the car is available for repair. This process may take several weeks, with the administrative overhead taking up most of the time. Additionally, many traditional insurers depend on paper applications, a practice that caused operational disruptions due to physical distancing during the coronavirus pandemic (Acharya, et al., 2020). While with today's technology it is already possible to completely eliminate the need for paper applications, the next step is to automate damage estimation and assessment. The damage estimate by the appraiser comprises different types of estimates: one for the damage to the different outer and inner vehicle parts, one for the residual value of the vehicle, and one for the replacement and repair costs. In the case of a *total economic damage* or *total technical damage* the insurer will pay out the replacement value of the car, deducing its residual value. A total economical damage is present, if the repair costs exceeds the replacement cost by a certain amount, usually 30%. In contrast a vehicle suffers a total technical damage, if no repair is possible. Since the residual value and repair costs of the vehicle influence the insurance payout, a damage evaluation of the vehicle is always necessary, even if the car is not redeemable (Verivox, n.d.). But less than critical accidents are also subject to some form of damage evaluation. Minor losses, such as scratches, can also be subject to an insurance claim. Although here, professional claims adjusters are only involved if the damage is close to safety critical areas, such as parking sensors. While the claims process itself is costly, involving repair and appraisal cost, the customer is also eligible for financial support of other services, such as car rentals, payments to the administration for unsubscribing and registering their car, etc. (Rhode, 2021)

Automation would significantly increase the customer experience, due to shortening the time span between accident and payout and/or repair. Customer experience, efficiency, and effectiveness are key performance indicators, that are known to generate significant value for insurance claim companies (Acharya, et al., 2020). McKinsey recommends to start by fully automating simple cases, allowing instant claims payouts as practiced by Lemonade's housing insurance (Sawers, 2017). While chat-bots may be sufficient to handle clear and simple cases, additional value can be generated by end-to-end digitization of the customer experience, reducing the administrative overhead. Moreover, analog damage appraisal and cost estimation will always form a bottleneck in the approval process. The difficulty of an automated damage assessment relying on computer vision is the context-sensitivity of the task. For example the damage evaluation of a dent in a door, is vastly different if the dent is near the edge or in the middle of its surface area, due to the different repair options applicable to the two damage types. Many major advances in computer vision in recent years have been realized with the help of deep learning. It excels in computer vision tasks where the semantic context of features plays a major role.

The goal of this paper is to train a deep learning system on simulated data for vehicle damage evaluation. Other systems in the damage evaluation domain do not rely exclusively on simulated data, but use real annotated data (Tractable Ltd., n.d.; Shaukat, 2019). The difficulty in using simulated data for deep learning lies in applying the system directly, without any re-training, to real world data. While previous works exist (Tobin, et al., 2017; Tremblay, et al., 2018) that address training deep learning systems exclusively on synthetic data, this work attempts to transfer damage, a quantity that is only measurable within simulations, to real world cases.
The available amount and quality of the training data is the key for successfully training a deep learning model. This project relies on simulated data for two reasons: $a)$ simulated data is, in comparison to real data, quick and cheap to obtain and $b)$ the privacy policy of the EU prevents companies seated in the European Union to train deep learning models on available customer data (Kop, 2020) – it is only allowed to use it for model evaluation.
While various options for simulators from the automotive domain are available, BeamNG GmbH stands out from the many providers of driving simulation software, through its detailed information on vehicle damage. Using a soft-body physics framework allows BeamNG.tech to easily compute in-depth damage statistics for individual vehicle parts. The subdivision of vehicles into individual parts, is also a feature that is advantageous to the project. For example the model chosen for this pilot project consists of 123 individual parts.
Thus, the contribution of this paper are twofold: First, an open source, and for academic research, free data set generation system for damage evaluation, which is based on the BeamNG.tech simulator. Second, a deep learning system for damage evaluation relying on synthetic data only.

1.2. Previous Work

Due to their low cost, using simulations for the training of intelligent systems gained more and more popularity in recent years. While many systems have been successfully trained on simulation data, a common problem is the transfer of learned heuristics to real-life data. This is commonly attributed to the differences between reality and simulation, also known as *reality gap*. One technique that has been crucial for improving real-to-sim transfer is *domain randomization*. With this method, the simulated world is configured in a way that introduces a high amount of variation into the data set for features that do not contain critical information needed to solve the given task. The goal is increasing the robustness of the deep learning system by forcing it to focus on relevant information. This idea was first explored by Tobin et al. (Tobin, et al., 2017) where the authors relied exclusively on simulated data to train a CNN (convolutional neural network) in an object location estimation task for robotic grasping. Using a fixed set of objects with random positions on a table, the authors varied their texture and color but also the position and orientation of the virtual camera generating the images. Their tests indicated, that a large number of textures is necessary for a successful sim-to-real transfer. Tremblay et al. (Tremblay, et al., 2018) continued to explore the idea of domain randomization in another work, where for the first time a deep learning system was trained exclusively on synthetic data and acquired an accuracy on a real-world data test set that is comparable to systems trained on real-world data only. Here, non-realistic images of vehicles were generated for a car detection task. Besides varying texture, color, etc., the researchers also introduced *flying distractors* – random geometry scattered throughout the scene.
Multiple companies offering (partially) automated damage evaluation and assessment are on the market. One of these being Tractable (Tractable Ltd., n.d.), which specialize in vehicle damage assessment and repair cost estimation. They rely on a computer vision system, trained on real data (Georgian, 2020) for their evaluation. Since deep learning systems require large amounts of data, its acquisition is one of the key factors in successfully training any deep learning based system. The lack of data for training is referred to as the *cold start problem*. Another solution relies on partial digitization of the claims process. The insurance company USAA relies on Mitchell International and Google Cloud for a semi-automated appraisal procedure (Shaukat, 2019). The customers digitally submit their claims application that includes images of the damage. A deep learning system then gives a preliminary estimate of the damage. The estimate is then re-evaluated by a professional claims adjuster. This feedback is in turn used as the final damage assessment and feedback (or ground truth) for improving the machine learning model. The deep learning system speeds up the claims process by assisting the claims adjuster and is in turn further improved by using incoming insurance claims as ground truth for further training.

1.3. Product Solution

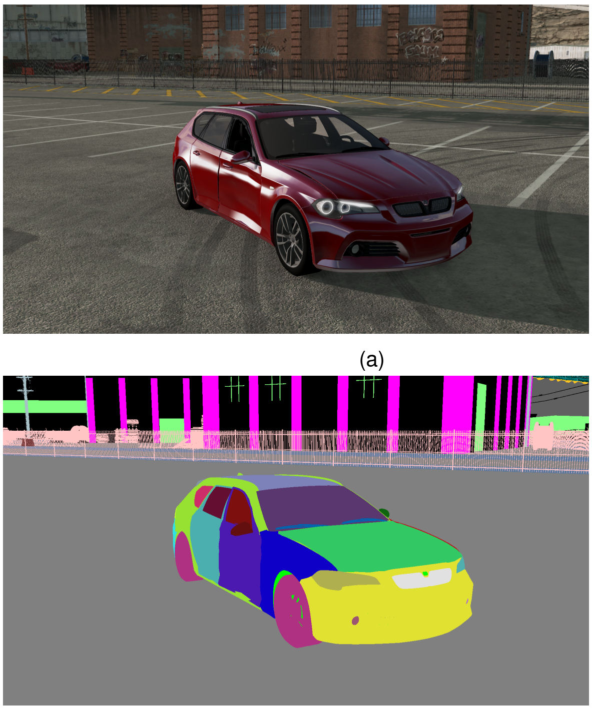
The two major components of this project are its data generation framework and training of a deep learning system on said data. This section gives a quick overview of the project components. For an in-depth discussion, the reader is referred to Section 2.

1.3.1. Impact Generator

BeamNG GmbH developed a custom data generation framework for image based damage detection. With the help of BeamNG.tech it simulates crashes and then uses the models of damaged vehicles to create images of detailed scenes that vary in different aspects, such as vehicle color, weather, time of day, and other environmental aspects. Four types of accidents were implemented: broadside collision, rear end, frontal impact, pole crash and no crash. For every image, a corresponding ground truth is generated that includes detailed metadata about the damage of the individual vehicle parts and a semantic annotation. See Figure 1 for example images. This annotation consists of an RGB image that assigns to every background object and vehicle part an individual color.

1.3.2. Computer Vision Pipeline Description

The computer vision setup consists of multiple deep learning models, that were trained on individual vehicle parts. To acquire the model architectures, a neural architecture search was performed using AutoKeras (Jin, Song, and Hu, 2019), resulting in an ResNet (He, et al., 2015) like architecture. To keep



(a)

(b)

Figure 1: Example output of the Impact Data Generator. (a) shows a vehicle damaged through a broadside collision, while (b) shows the corresponding semantic segmentation.

the computer vision pipeline compatible with the production environment, where the vehicle parts hierarchy may change, individual models were trained on specific vehicle parts. This required a filtering of the available training data according to visibility and received damage of a given vehicle part. The deep learning systems then learned to predict damage, or no damage for their respective vehicle parts.

2. Methodology

2.1. Data Generation Framework

2.1.1. Soft-Body Dynamics

To understand how damage is computed in BeamNG.tech a deeper understanding of the kinematics model is needed. The soft-body dynamics framework BeamNG.tech relies on, is based on a spring-mass model, where every object in the simulation consists of a 3D graph. In the BeamNG.tech terminology, springs are called *beams* and mass points are called *nodes*. Nodes have a position in space that is updated dependent on the net forces acting on them, and also influences the state of the beams. BeamNG.tech beams are similar to beams in mechanical engineering, which are structural elements that withstand load primarily by resisting bending: They have a length that may change depending on the force acting on it, i.e. depending on the velocity of the nodes. The change in length of a beam will from here on be called deflection. Deflection can be *elastic*, i.e. the beam will be restored to its original rest length in the absence of force, or *plastic*, i.e. the deformation will change the rest length of the beam, and it can break the beam. Broken beams no longer influence the position of the nodes they are connecting. Beams have deformation and break thresholds that are used to mimic the properties of different materials. The deformation threshold determines how much force can act on the beam before it shows a plastic (i.e. non-reversible) deformation. In turn the break threshold determines how much force can act on the beam before it breaks. To mimic the properties of a sponge-like material, one would set a low deformation threshold and a high break thresholds, since sponges are easy to deform but difficult to break. To model a brittle material like glass one would choose a deformation threshold that is higher then the break threshold to avoid a deformation of the beam before it breaks.

2.1.2. Damage Computation

The damage computation used in this project is closely related to the spring-mass model. While many more values are available through the simulation, the main factor used to determine whether a vehicle part shows some form of damage, was the maximal beam deformation ratio in terms of length of a vehicle part. The deformation ratio of a beam is the ratio between its length at rest and current length, independent of whether the beam has been compressed or extended. The strain of a beam is its change in length relative to its original length. Thus, the measure relies on the maximum deformation value of all the beams that belong to the same vehicle part.

2.1.3. Data Generation

For the data generation BeamNG GmbH developed an Impact Data Generator that is available as an open source project [†]. It is a Python package with a custom extension for BeamNG.tech. While the Python side manages the BeamNG.tech process, scenario definition and data management, the Lua side is responsible for executing the scenarios. The output data consists of the deformed vehicle model (the term model refers here to the visual model with the 3D physics graph, not the physics model used to simulate vehicle motion), images of the vehicle after the accident together with the ground truth. The ground truth consists of a semantic annotation with an individual color for the different car parts and background objects. For each scenario an image series showing a 360 degree surrounding view of the vehicle were generated.

2.2. Data Set Description

For the data set generation BeamNG.tech's Impact Data Generator was used. A total of 4.000 scenarios were defined through domain randomization. The word scenario hereby refers to the setup of the simulation with a specific set of parameters (accident type, vehicle color, weather, etc.). For the scenario generation a random combination of the vehicle configuration (color, variants of different vehicle parts such as bumper, fender, hitch, etc.), and environmental settings (time of the day, weather) were selected. Four types of accidents are available through the Impact Data Generator, some involving only one vehicle,

[†] https://github.com/BeamNG/ImpactGen

some involving two vehicles: broadside collision, rear end, frontal impact, pole crash, and no crash. Every scenario was generated with the ETK800, a fictional vehicle model included in BeamNG.tech. Through this process 2.5TB of data were acquired.

2.3. Model Description

The architecture was selected with the help of AutoKeras (Jin, Song, and Hu, 2019) (version $0.4.0$ with a Pytorch backend), a neural architecture search framework. AutoKeras automatically selects the most suitable model it can find in its search space through *network morphism*. Network morphism allows the framework to change the architecture of the model while keeping its functionality and requiring only few epochs of training for improving performance.

The resulting model architecture was fine tuned in a subsequent step, where a more exhaustive hyperparameter search was performed with more available training data. This resulted in a ResNet (He, et al., 2015) like model. For this pilot project two models were then trained – one to detect bumper damage and one to detect damage on the fenders. The reason for training models on individual car parts, is to be able to quickly change the production setup to match any changes in the vehicle parts classification system. The input data consisted of downsized images with 300x300 pixels, while the output consisted of the model's prediction on whether the respective vehicle part was damaged or not. Training the network on images containing a specific vehicle parts required filtering the data set by a *data provisioning pipeline*. This custom tool filtered the images according to visibility of the vehicle part in question and image angle. The tool also controlled the portion of images in the data set showing the respective damaged vs undamaged vehicle part. Similar to the method of Tremblay et. al (Tremblay, et al., 2018), the simulated data was further augmented by combining the foreground of the synthetic images with random backgrounds. For the training the network was initialized with random weights and then trained using stochastic gradient descent and mean squared error.

2.4. Performance

The network achieved an accuracy of 94% on the synthetic evaluation set. But evaluating the network's performance on real-world data, resulted in a dramatic drop of the accuracy to 46%. It is noted that the real-world images of damaged vehicles for the evaluation originated not from customer data, but are images that were taken by the team members of their own vehicles to comply with the EU privacy policy.

3. Discussion

3.1. Results Evaluation

The limited transfer capabilities of the model from synthetic data to real world data may have multiple reasons. For one, the provided data may show too little variation in comparison to real world images, which could be countered by implementing more domain randomization strategies. Considering the work of Tremblay et. al (Tremblay, et al., 2018), one option is to make use of flying distractors that may occlude the vehicle. We observed that vehicle occlusion dramatically impacted the model performance, although it is a commonly encountered property of real world data.

3.2. Impact of the proposed solution

As stated by McKinsey, automatic damage evaluation is a necessity for any competitor in the insurance claims industry (Acharya, et al., 2020). It contributes to increase consumer convenience by augmenting service quality. Due to the EU privacy policy restrictions discussed in Section 1, simulated data is the only possibility for EU-based companies to deploy custom deep learning systems without relying on third party providers.

4. Conclusion

In summary this preliminary work presents the benefits and challenges of sim-to-real transfer as it pertains to damage estimation. Options in exploring future possibilities of this approach have been discussed in the previous section. Because real data is difficult to acquire and even more expensive to annotate, simulations present a feasible alternative.

References

Acharya, A., Hamilton, D., Patiath, P., Surak, Z., Dienstag, G. T., and Ouwerkerk, J. van, Apr. 2020. *Insurance resilience in a rapidly changing coronavirus world | McKinsey*. Available at: ¡https://www.mckinsey.com/industries/financial-services/our-insights/insurance-resilience-in-a-rapidly-changing-coronavirus-world¿ [Accessed Mar. 31, 2021].

Georgian, May 2020. *AI for Accident and Disaster Recovery Has Arrived with Tractable CEO, Alex Dalyac*. en. https://growth.georgian.io/c/preview?x=81x44X&utm_source=linkedin.

He, K., Zhang, X., Ren, S., and Sun, J., Dec. 2015. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*. arXiv: 1512.03385. Available at: ¡http://arxiv.org/abs/1512.03385¿ [Accessed Apr. 9, 2021].

Jin, H., Song, Q., and Hu, X., Mar. 2019. Auto-Keras: An Efficient Neural Architecture Search System. *arXiv:1806.10282 [cs, stat]*. arXiv: 1806.10282. Available at: ¡http://arxiv.org/abs/1806.10282¿ [Accessed Mar. 30, 2021].

Kop, M., Mar. 2020. Machine Learning & EU Data Sharing Practices. en, p. 16.

Rhode, S., Feb. 2021. *Wirtschaftlicher Totalschaden am Pkw: Wer zahlt wann und wie viel?* Available at: ¡https://www.check24.de/kfz-versicherung/schaeden/totalschaden/¿ [Accessed Mar. 31, 2021].

Sawers, P., Dec. 2017. *Lemonade raises $120 million from SoftBank, others to take its chatbot-based insurance service global*. en-US. Available at: ¡https://venturebeat.com/2017/12/19/lemonade-raises-120-million-from-softbank-others-to-take-its-chatbot-based-insurance-service-global/¿ [Accessed Mar. 29, 2021].

Shaukat, T., Sept. 2019. *USAA and Google Cloud work together to speed auto claims*. en. Available at: ¡https://cloud.google.com/blog/topics/customers/usaa-and-google-cloud-work-together-to-speed-auto-claims/¿ [Accessed Mar. 31, 2021].

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P., Mar. 2017. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. *arXiv:1703.06907 [cs]*. arXiv: 1703.06907. Available at: ¡http://arxiv.org/abs/1703.06907¿ [Accessed Mar. 29, 2021].

Tractable Ltd., n.d. *Artificial intelligence for accident & disaster recovery | Tractable*. en-GB. Available at: ¡https://tractable.ai/¿ [Accessed Mar. 29, 2021].

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., and Birchfield, S., June 2018. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Salt Lake City, UT: IEEE, pp. 1082–10828. ISBN: 978-1-5386-6100-0. https://doi.org/10.1109/CVPRW.2018.00143. Available at: ¡https://ieeexplore.ieee.org/document/8575297/¿ [Accessed Jan. 26, 2021].

Verivox, n.d. *Wirtschaftlicher Kfz-Totalschaden: Das muessen Sie wissen*. Available at: ¡https://www.verivox.de/kfz-versicherung/themen/wirtschaftlicher-kfz-totalschaden/¿ [Accessed Mar. 31, 2021].