Automation of Leasing Vehicle Return Assessment Using Deep Learning Models

Mohsan Jameel[1](✉), Mofassir ul Islam Arif[1], Andre Hintsches[2], and Lars Schmidt-Thieme[1]

[1] Information Systems and Machine Learning Lab, University of Hildesheim, Hildesheim, Germany
{mohsan.jameel,mofassir,schmidt-thieme}@ismll.uni-hildesheim.de
[2] Volkswagen Financial Services AG, Braunschweig, Germany
Andre.Hintsches@vwfs.com

Abstract. The vehicle damage assessment includes classifying damage and estimating its repair cost and is an essential process in vehicle leasing and insurance industries. It contributes heavily to the actual cost the customer has to pay. The standard practices follow manual identification of damages and cost estimation of repairs, resulting in noisy images of the damaged parts, inconsistent categorization of damage types, and high variance in repair costs estimation between two appraisers.

We employ explainable machine learning to highlight how the standard ML models and their training protocols fail when dealing with a dataset acquired without a standard procedure. In this paper, we present a multi-task image regression model for the leasing vehicle return assessment that leverages the car configuration to reduce the cost of repair assessment. Our solution achieves a 50% error reduction in the repair cost estimates. Furthermore, we present remedies base on hierarchical taxonomy and cost-sensitive loss to improve the damage classification accuracy.

Keywords: Image classification · Computer vision · Cost-sensitive · Deep learning · Explainable machine learning

1 Introduction

Leasing vehicles such as luxury cars, cooperate vehicle fleets etc., is an attractive option for many customers as it provides a cost-effective alternative to buying those vehicles. It is estimated that the market share of the leasing vehicle industry will grow more than USD 300 billion by 2021 [1]. The vehicle is used by the customer for a contracted period of time. At the end of the contract, an appraiser inspects the vehicle for damages and generates a report using the pictures of damages and their associated repair cost. Traditional methods rely on manual identification of damage and cost estimation of repairs, which results

M. Jameel and M. I. Arif—Both authors contributed equally to this research.

Y. Dong et al. (Eds.): ECML PKDD 2020, LNAI 12460, pp. 259–274, 2021.
https://doi.org/10.1007/978-3-030-67667-4_16

in noisy images of the damaged parts, inconsistent categorization of damage types, and high variance in repair costs estimation between two appraisers. The high variance in the cost of repair means that either the customer or the leasing company were overburdened by the disproportionate estimates.

In recent years, the enhancement in the modeling capacity of deep learning models for image analysis have made the automation efforts feasible in many fields such as medical image diagnostics, roadside sign recognition, autonomous driving, predictive maintenance, etc. Damage assessment of leased vehicles presents another challenging application with huge potential to reap benefits of advancement in the area deep learning and computer vision. The damage assessment comprises two main components, 1) identification and classification of the damage type and 2) predicting the cost of repair for that particular damage. The two components are related, as the accurate classification leads to accurate cost estimates. Although, there are many off-the-shelf deep learning solutions for object detection and classification, however, tuning them to an industrial setting brings its own challenges.

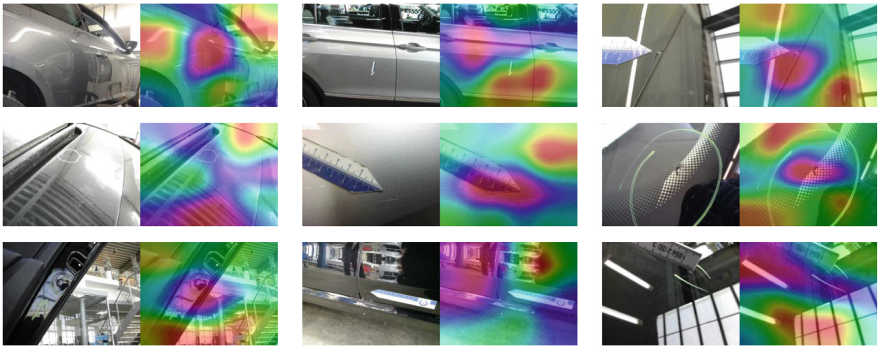

(a) Dent (b) Scratch (c) Stone chip

Fig. 1. In each sub figure, the image on the left is the original and on the right is GradCAM generated overlay. The region of the images used by the model for decision making are highlighted using GradCAM. The magenta color region surrounded by violet is the focus region used for decision making. (Color figure online)

In this paper, we used the data collected by one of the leading vehicle leasing company in Europe. The data collected by the traditional approach was highly noisy, unstructured and labels were inconsistently categorized as in contrast to benchmark datasets available for the research purposes. To showcase the problem, we trained an *Inception_v3* model using transfer learning techniques and fine-tuned it on the label images from the company data. The classification model was able to predict the correct damage labels with a nominal 50% accuracy. The strength of these models comes from extracting useful representations from images, these representations are then used for decision making in a classification setting. To further investigate the reason for low accuracy, we employed

GradCAM [10] to explain the decision made by the model, which is presented in Fig. 1. Although there are some instances for which the model was able to base its decision on the correct damage representation, however, due to the high level of noise and incorrect images, a large majority of the decisions are based on incorrect representations such as classifying a 'Scratch' based on the floor of the workshop. Another main issue was the incorrect categorization of the damages, for example, damages with similar visual representation were given two different labels. Lastly, the cost of repair estimates has a high variance between different observations for similar damage types.

In this paper, we tackle these problems using a combination of well-established pre-processing techniques and explainable machine learning to identify and rectify the problems in the automation process. Firstly, we properly annotated the images using bounding boxes that help in capturing a proper representation of the damage types and remove noisy images. The problem of inconsistent damage labels was tackled by defining the hierarchical class taxonomy. Secondly, to better utilize the cost of repair information in the damage classification, we defined a cost-sensitive classification loss. And Lastly, we define a cost regression model that uses both images and vehicle meta-features to predict the cost of repair.

To recap, our contributions are:

- We used a data-driven pre-processing procedure for adapting an industrial dataset to a machine learning problem and used explainable machine learning to define better damage categorization.
- We define a cost-sensitive classification loss as the classification error has an associated penalty in terms of cost estimation.
- We present a cost regression model that leverages both car information as well as damage images to reduce the variance in cost estimation.

2 Related Work

The detection and classification of damage from the picture and assessing the cost is the main task of the leasing vehicle return assessment process. The assessment of damage is not unique to the leasing vehicle return assessment process. It is a core component of the insurance claim process such as vehicle and housing damage claims. However, there are limited research studies conducted in this particular area. In this direction, Patil et al. [8] created a small dataset of damage cars through web crawling. They used some standard CNN models to extract image features and feed it to an SVM classifier for predictions. The dataset is limited to only dent and broken glass/light damage types and did not include any cost estimations. Li et al. [5] conducted a study on detecting the fraud in a car insurance claim and generated a damage dataset by crawling the Internet for the damage images. They used an object detector to identify damage parts and build a system to check for fraudulent claims. Although both the studies target detection of the damages but they still fall short of providing or discussing a complete solution for damage assessment. Previous studies were limited in their scope of exploring other types of damages that are frequent in the real world dataset. On the contrary they focused on dent and scratch, which are easily distinguishable due to distant features. In our case study, we worked with 14 different damage types, which occur frequently in real-world applications.

There are some literature available on related applications on damage detection. Maeda et al. [6] conducted a study on detection of road damage such as cracks. The data was collected using mobile device, which consists of 8 different types of damages, and used variety of object detection models to build an automated solution. Similarly, the assessment of damages to a building after disaster was studied by [7]. There is a commercial interest in the automation of the damage insurance claim, which is evident from the fact that there are number of startups working in this area such as Ant Financial and Tractable.ai to name a few.

3 Methodology

In this section, we will formulate the leasing vehicle assessment process as a multi-task machine learning problem and present the cost-sensitive loss for damage classification.

3.1 Problem Formulation

The leasing vehicle return assessment process consists of two main tasks, i.e classify a damage type and estimate its cost of repair. Generally, a multi-task learning [11] setup best suits this type of problem. Let $\mathcal{X} = \{\mathcal{X}^v, \mathcal{X}^p\}$ define a set of input space, where $\mathcal{X}^v \in \mathbb{R}^V$ is a set of vehicle features such as model, make, color, body part, etc., and $\mathcal{X}^p \in \mathbb{R}^{H \times W}$ is a set of associated pictures/images to a capture visual representation of specific damage. The task-specific output space $\mathcal{Y} = \{\mathcal{Y}^d, \mathcal{Y}^c\}$, where $\mathcal{Y}^d \in \mathbb{R}^D$ represents a set of damages and $\mathcal{Y}^c \in \mathbb{R}$ represents the cost of repair. The dataset set $\mathcal{D} = \{\mathbf{x}_i, y_i^d, y_i^c\}_{i=1}^N$ consists of N observations. To learn a joint model for two tasks, we have two sets of model parameters, a set of model parameters θ^s that is shared between tasks and task specific model parameters θ^d and θ^c. We want to learn a mapping function for each task, which can be defined as,

$$\hat{y}_d(\mathbf{x}, \theta^s, \theta^d) : \mathcal{X} \rightarrow \mathcal{Y}^d \tag{1}$$

$$\hat{y}_c(\mathbf{x}, \theta^s, \theta^c) : \mathcal{X} \rightarrow \mathcal{Y}^c \tag{2}$$

We also have a specific loss for each task i.e. a cross-entropy loss $\hat{\mathcal{L}}_d(\cdot, \cdot)$ for damage classification and squared loss $\hat{\mathcal{L}}_c(\cdot, \cdot)$ for cost of repair assessment. The multi-task objective function thus becomes:

$$\underset{\theta^s, \theta^d, \theta^c}{\arg\min}\, \alpha^c \hat{\mathcal{L}}_c\left(y^c, \hat{y}_c(\mathbf{x}, \theta^s, \theta^c)\right) + \alpha^d \hat{\mathcal{L}}_d\left(y^d, \hat{y}_d(\mathbf{x}, \theta^s, \theta^d)\right) \tag{3}$$

where $\hat{\mathcal{L}}^j(y^j, \hat{y}_j(\mathbf{x}, \theta^s, \theta^j)) = \frac{1}{N}\sum_{(\mathbf{x},y^j)\in\mathcal{D}} \mathcal{L}_j(y^j, \hat{y}_j(\mathbf{x}, \theta^s, \theta^j)), j = \{d, c\}$. The task-specific weights $\alpha^c \in \mathbb{R}^+$ and $\alpha^d \in \mathbb{R}^+$ are hyperparameters, which are used to control the weight of a specific task in the overall loss. However since we are dealing with an industrial dataset, the images are noisy and collected without a machine learning application in mind. Therefore, directly using a machine learning model on this dataset does not yield the desired results. With this in mind, we propose to solve the classification and regression problem separately.

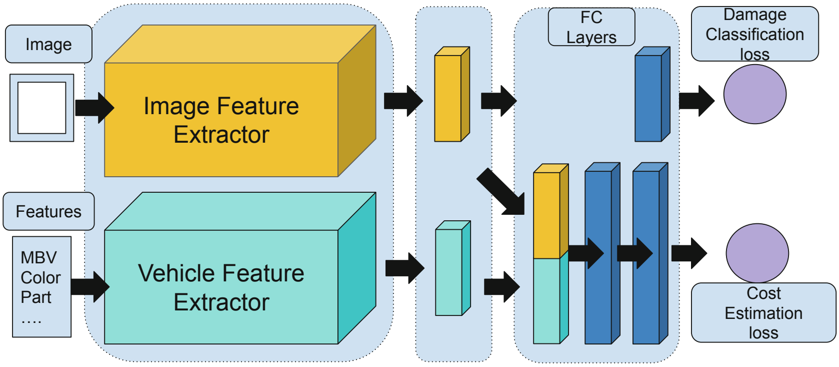


Fig. 2. The cost regression model for vehicle leasing return assessment process. The upper part of the diagram corresponds to the image classification model and lower part corresponds to a cost regression model.

3.2 Damage Classification

Damage classification is an important part of cost of repair estimation since the type of damage directly impacts the cost. We have used *Inception_v3* and *Resnet20* as image feature extractors and since they are complex models we have used transfer learning to initialize their weights pre-trained on ImageNet. Transfer learning has shown to be an effective method to retrain a model with limited data. For training the models, we propose to use two variations of the multi-class classification loss function $\mathcal{L}_d(\cdot, \cdot)$, the standard cross entropy loss and cost-sensitive classification loss.

Cross Entropy Loss: The cross entropy loss is used as a proxy loss for a misclassification rate, defined in Eq. (4).

$$\mathcal{L}_d\left(y^d, \hat{y}_d(\mathbf{x}, \theta^s, \theta^d)\right) = \begin{cases} 1, & \text{if } y^d \neq \hat{y}_d(\mathbf{x}, \theta^s, \theta^d). \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

In this loss function, if a model prediction does not match the target label, it incurs an error. The error is always one, irrespective of the incorrect label selected by the model. This loss is widely used and best suited for situations in which the penalty for all misclassifications is equal. The cost matrix for misclassification is given in Table 1. For example, if the model misclassifies a scratch as a dent or stone-chip, the penalty of the mistake is the same.

Table 1. (left) Cost matrix for misclassification, and (right) Cost matrix based on average cost difference between pairs of damages

Damage Class	Scratch	Dent	Stone-chip	Wear	Burnt-Hole
Scratch	0	1	1	1	1
Dent	1	0	1	1	1
Stone-chip	1	1	0	1	1
Wear	1	1	1	0	1
Burnt-Hole	1	1	1	1	0

Damage Class	Scratch	Dent	Stone-chip	Wear	Burnt-Hole
Scratch	0	49	327	28	34
Dent	49	0	377	78	84
Stone-chip	327	377	0	298	292
Wear	28	78	298	0	5
Burnt-Hole	34	84	292	5	0

Cost-Sensitive Classification: In many applications, the cost for misclassification is not the same for all types of mistakes, for example, customer churn prediction. In our problem, we are given the cost of repair for each instance of the damage. The cost of repair of two different damage types could vary significantly. For this purpose we created a cost matrix by recording for each pair of damage the difference between their average cost of repair, a subset of the cost matrix is shown in Table 1. Again taking the same example as before, now if a scratch is misclassified as a dent, it will incur a penalty of 49. On the other-hand, misclassifying a scratch as a stone-chip will result in a penalty of 327. Therefore, we used this information in the loss function and define a cost-sensitive loss given in Eq. (5).

$$\mathcal{L}_d\left(y^d, \hat{y}_d(\mathbf{x}, \theta^s, \theta^d)\right) = \begin{cases} c_{y^d, \hat{y}_d(\mathbf{x}, \theta^s, \theta^d)}, & \text{if } y^d \neq \hat{y}_d(\mathbf{x}, \theta^s, \theta^d). \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where $c_{\cdot,\cdot}$ is an element of the cost matrix $\mathbf{C} \in \mathbb{R}^{D \times D}$.

3.3 Cost Regression

The task of predicting the cost of repair can be categorized as a regression problem, which is defined in Eq. (2). There are many state-of-the-art machine learning models, such as Gradient Boosted Decision Trees (XGB) [3] and Random Forest (RF) [2], which have shown to perform exceptionally good on the regression task for vector data. However, in our problem, we were given a mix of vector data and images, more specifically, damage labels are encoded in images. To include both, the vector data such as car information and pictures of damage, we use a deep neural network. We used a CNN based feature extractor i.e. *Inception_v3* and *Resnet*, to learn the latent representation of the images. The latent representations of images are concatenated with car features and become the input to the fully connected feed-forward neural network, as shown in Fig. 2. We used mean squared error (MSE) loss $\mathcal{L}_c(y^c, \hat{y}_c(\mathbf{x}, \theta^s, \theta^c) = (y^c - \hat{y}_c(\mathbf{x}, \theta^s, \theta^c)^2$ to train the model.

$$\hat{\mathcal{L}}_c\left(y^c, \hat{y}_c(\mathbf{x}, \theta^s, \theta^c)\right) = \frac{1}{N} \sum_{(\mathbf{x}, y^c) \in \mathcal{D}} (y^c - \hat{y}_c(\mathbf{x}, \theta^s, \theta^c)^2 \tag{6}$$

4 Experiments

This section talks about the dataset, the steps taken to make is compatible with a machine learning setting and lays out the results for our classification and cost regression.

Table 2. Statistics of leasing vehicle return dataset

Name	Reports	Images with Cost	Damage Types	Models (mbv)	Colors	Parts	repair actions
Count	39,000	342,029	35	51	165	166	21

Table 3. Statistics of dataset after annotation phase

Name	Damage classes	Sampled Images	Annotated Images	Total Crops
Count	14 + 1	48,000	17,083	25,228

4.1 Dataset

The dataset used in this paper was collected by one of the leading vehicle leasing company in Europe. It is made up of $40,000$ reports that have been generated manually by appraisers at the end of a leasing contract. The appraiser inspects the car for damages, identify the damages, photograph them, and provides an estimate for the cost of those repairs. There are $342,029$ photograph images of damages and each image has a corresponding body part, damage type and the estimate for the cost of repair. Overall, there are 166 meta-level body parts and 35 damage types available in the collected dataset. Apart from the damage specific information, we also have detailed meta-features about the vehicle out of which the more relevant features are model, make and color. The information of the car model was available at a very fine grain level i.e. interior configurations, and variation in the trim levels. We combined these models in high-level groups represented by MBV, which are based on the model rather than the variants of the same model, for example, the same car model with different trim levels is treated as one model. The color of the car also plays an important role in the repair cost estimation, as metallic or exotic colors cost more than the standard colors. Table 2 provides an overview of the number of reports in the dataset and the final number of these features.

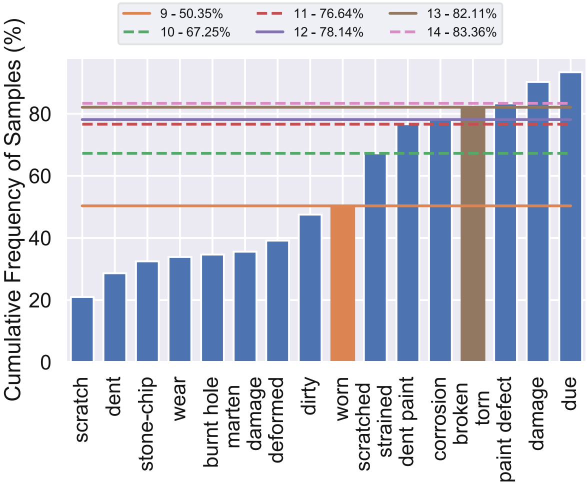
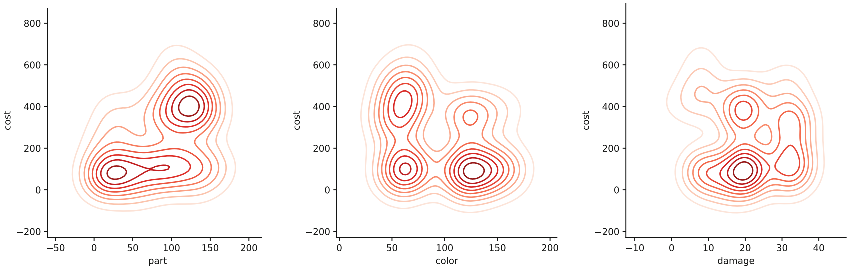


Fig. 3. The frequency of samples for top-14 damage types, represented as a cumulative frequency plot. It shows that first 9 classes listed from left to right covers around 50% of the data, whereas, top 14 classes covers 83% data. This shows only 14 classes out of 35 total classes constitute majority of the data.

The dataset consists of 35 damage types, however, there are two main problems with these damage types. Firstly, the damage types can be categorized into optical and non-optical damages. Optical damage has a visual appearance and can be captured through pictures, for example, 'dent', 'scratch' and 'stone-chips' etc. Whereas, non-optical damage cannot be captured or defined using visual features, for example 'smell of a bad odor', 'missing item' or 'play in a component'. Therefore, the non-optical damage types cannot be included in the classification task. Secondly, the damage types suffer from a typical long-tail distribution, some of the damage types did not have enough samples. To overcome these two problems, we picked 14 most frequent optical damage classes, which are shown in Fig. 3 as a cumulative frequency bar plot. We can see that ≈84% of the dataset can be covered using only the top 14 classes.

4.2 Exploratory Data Analysis

In this section, we perform an exploratory data analysis to understand the useful relationship between different features. We used Kernel Density estimate of cost and different features and plotted them in Fig. 4[1]. To presents more meaningful information in these plots, we used a single car model (mbv) to represent the relationship between the color, damage, part, and the cost. In Fig. 4(a) shows the cost of repair of a particular body part is higher than other, which verifies that different body parts require a different type of repairs. Figure 4(b) shows the cost relationship with color and again some colors have a higher cost of repair. It is also to be noted that it might also depend on the extent of the damage i.e. a small scratch might cost less to repair than a bigger scratch. Lastly, Fig. 4(c) shows the relationship between the cost and damage, which is similar to the color relationship. This can be caused by the extent of that damage but it is highlighted that the final cost for damage is also impacted by the variance in the opinion of an appraiser. We also wanted to see how the different parts and colors were related to the damages, to see if particular damage is always related to a certain part/color. Figure 5(a) shows that damage and color do not hold a strong correlation as is expected. Conversely, we can see in Fig. 5(b) that the damage and part appear to have a strong correlation. A 'stone chip' frequently appears at the curved lining, where the paint is weakest. From this analysis, we are able to infer that the model, color, part and damage under consideration have an impact on the final cost and therefore need to be included in the model as auxiliary information.



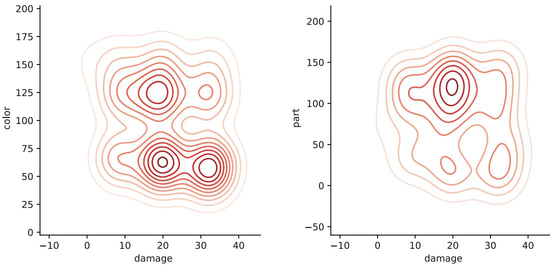
(a) cost vs parts (b) cost vs color (c) cost vs damage

Fig. 4. The plots represents Kernel Density estimate between cost and different car features present in the dataset.

4.3 Data Cleaning and Annotation

The task of image classification relies heavily on the quality of the images being trained on. The damage images in the reports are taken without a standard

[1] The values of the cost of repair is always greater than 0, however, because of the kernel density function some contours appear to be below zero values.



(a) damage vs color (b) damage vs parts

Fig. 5. The plots represents Kernel Density estimate between damage and other car features present in the dataset.

acquisition procedure and therefore vary significantly. Variance in lighting conditions, distance from the damage, noisy backgrounds, and even dirty car parts make the task of learning useful representations more challenging. In order to learn a useful classifier for the damages, we annotated the dataset using bounding boxes. We annotated the images with bounding boxes and marked those images as ‘dirty’, which have a noisy background, dirty car, poor lighting, high reflections, and blurrey images. We randomly sampled 3500 images from each damage class to be annotated but because of the high level of noise, only 17, 083 were annotated, while the rest were marked as ‘dirty’. Furthermore, we created crops of images using the bounding boxes, which resulted in 25, 000 crops of damages. These crops are useful to learn a damage classifier, as crops capture the visual representation of damages while reducing the background noise. The summary of the statistics for the bounding box annotations are presented in Table 3 and Fig. 7. Examples of the crops generated by the annotation phase are presented in Fig. 6. An extra class was included, which we called a ‘negative damage class’ to provide negative examples for training.

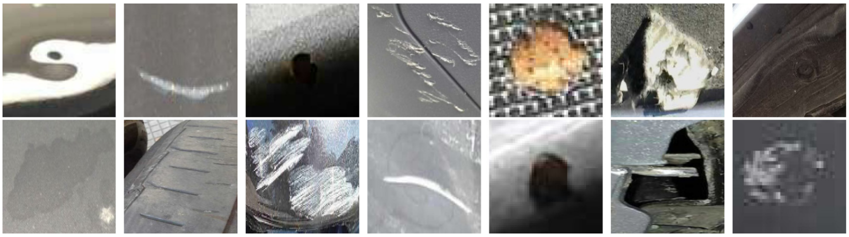


Fig. 6. Crops of damages

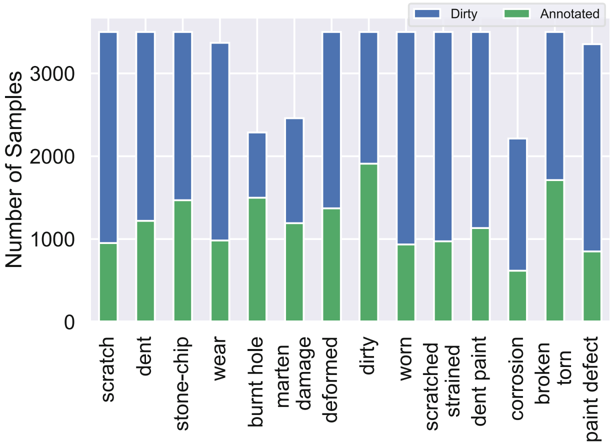


Fig. 7. The number of samples for annotation vs actual clean images annotated.

4.4 Damage Classification

In this section, we perform the experiments for the classification task. For this purpose, we used 25, 228 crop images dataset, which consists of color images, sized 225×225, with 14 damage classes and a 'negative damage class'. The data was split into 90% train-set and 10% test-set, such that test-set contains equal samples from each damage class. We perform 10 experiment runs, and for each run creating a new train/test split. We used *Inception_v3* [12] and *Resnet20* [4] pretrained on the ImageNet dataset [9]. The training of these models was done using SGD with momentum $\mu = 0.99$ and the learning rate η was searched in the grid $\eta = \{0.001, 0.01, 0.05, 0.1\}$.

Cross Entropy Loss. In the first set of experiments, we trained the classifier using standard cross entropy loss given in Eq. (4). The results presented in Fig. 10(a) show the classification accuracy on varying the number of damage classes. It is evident from the results, as we increase the number of classes, the complexity of the problem increases and the accuracy drops. The first column for 3 damage classes consists of 'Scratch', 'Dent' and 'Chip-Stone', which have very distinctive damage patterns, therefore, both the models were able to achieve very good results. However, once we start to increase the number of classes, the accuracy starts to degrade. The most significant drop in accuracy was observed at 10 classes and more.

In order to investigate the performance degradation, we used an explainable machine learning approach called GradCAM [10], which provides a method to visualize the gradients of the image per pixel and gain insight on the regions in an image used by the model for its decision to assign a particular class. The GradCAM analysis on a few similar classes is shown in Fig. 8. At a cursory glance, it becomes evident that the images for 'Scratch' and 'Scratched' classes,

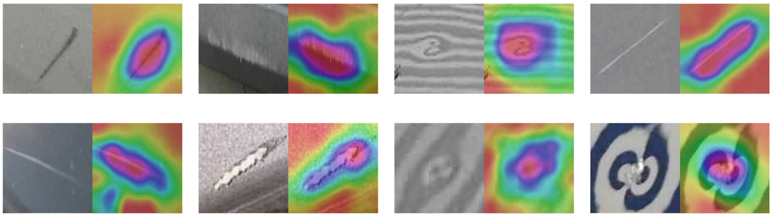
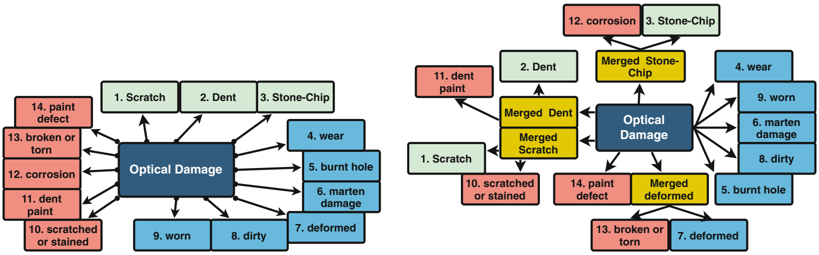

(a) Scratch (b) Scratched (c) Dent (d) Dent Paint

Fig. 8. Analysis of the damage crops using GradCAM. It highlights that some damage types are visually similar.

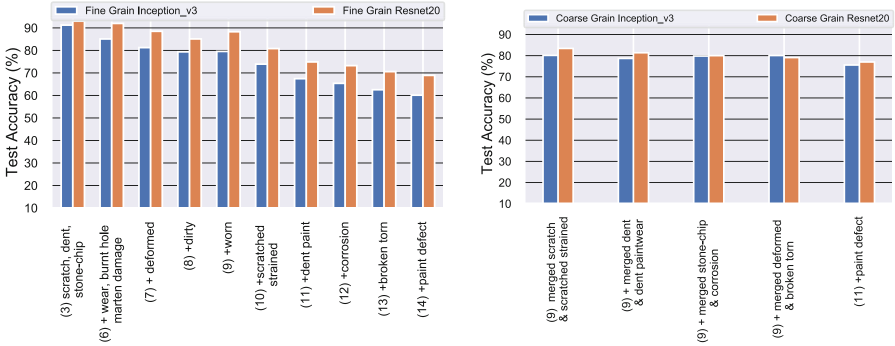
and 'Dent' and 'Dent Paint' appear to be causing very similar activations in the model. This is caused by the similar manifestation of the damages on the car i.e 'Dent' and 'Dent Paint' are both visually similar. This will lead to confusion between these classes and lead to poor classification accuracy.



(a) Fine Grain (b) Coarse Grain

Fig. 9. Damage class taxonomy, a) original taxonomy and b) proposed taxonomy

In addition to GradCAM, we analyzed the confusion matrix on our test set to identify the confusing cases. A certain pair of classes are being confused with each other, which was evident from the confusion matrix, for example 'Dent Paint' and 'Dent', and 'Corrosion' and 'Stone chip' are frequently confused. This problem highlights that the degradation in the performance of a machine learning model is not necessarily caused by the training or model choice, but it stems from the non-standard categorization of the damage labels. To rectify the problem of non-standard categorization of the damage labels, we proposed to group similar classes based on their visual representations. We defined a hierarchy taxonomy of the damage labels, which we referred to as 'Coarse Grain' (CG) taxonomy Fig. 9(b), whereas, the original class taxonomy is referred as 'Fine Grain' (FG) taxonomy Fig. 9(a). The classification accuracy for the CG taxonomy is presented in Fig. 10(b). It is observed that both *Inception_v3* and *Resnet20* model perform at par with each other. To compare the results of FG and CG taxonomy, we have to compare 9 classes results in Fig. 10(a) with Fig. 10(b), and it becomes clear that despite increasing the confusing samples by keeping the number of classes same, there is no degradation in the accuracy.



(a) Fine Grain

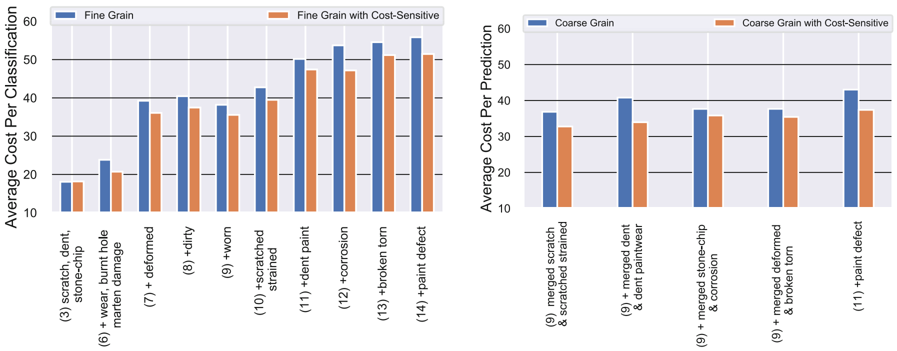
(b) Coarse Grain

Fig. 10. The classification accuracy on the test-set was presented a) for Fine Grain taxonomy and b) for Coarse Grain taxonomy. The numbers in the () on the x-axis represents the case of the number of classes and '+' sign represents that these classes are added to the classes already present in the left bar.

Cost Sensitive Classification: In the second set of experiments, we trained the damage classifier using cost sensitive loss given in Eq. (5). We used the same training protocol as in the previous section, the only change was the evaluation metric, which is changed from accuracy to cost-sensitive cost define similar to Eq. (5). The results are presented in Fig. 11(a) and Fig. 11(b) for FG and CG taxonomies respectively. The models trained on cost-sensitive loss had a lower misclassification error as compared to the one which was trained on the misclassification rate. It is also evident from the results if the problem is well defined, for example in the case of classification of 3 damage types, the misclassification error is very low, therefore, the performance of both the methods is equal.

4.5 Cost Regression

In this section, we perform the experiments for the prediction of the cost of repair. We used the same dataset as explained in the damage classification section, however, now the target is to predict the cost of repair. We used the car features given in Table 2 with 14 damage classes to predict the cost of repair. The data was split using a three-fold validation strategy, where two folds are used for training and one for testing. The state-of-the-art models such as RF and XGB were trained on this data excluding the images and using the appraiser assigned damage type. We also build a custom Feed Forward neural network (FNN), which consists of two fully connected layers with Relu activation function and



(a) Fine Grain

(b) Coarse Grain

Fig. 11. The average cost of misclassification (lower the better) on the test-set was presented a) for Fine Grain taxonomy and b) for Coarse Grain taxonomy. The numbers in the () on the x-axis represents the case of the number of classes and '+' sign represents that these classes are added to the classes already present in the left bar.

Table 4. The results of cost regression task.

Model	Features	RMSE
Average model	∅	237.43 ± 0.73
Linear regression	Parts, mbv, color, damage, action	106.41 ± 3.81
Random Forest (RF) [2]	Parts, mbv, color, damage, action	85.84 ± 3.99
XGboost (XGB) [3]	Parts, mbv, color, damage, action	84.77 ± 1.78
FNN (our)	Parts, mbv, color, damage, action	**82.3 ± 2.8**
FNN + Image (our)	Image, parts, mbv, color, action	**83.6 ± 0.73**

dropouts. We performed extensive grid search to find the optimal number of nodes $\{32, 64, 128, 256\}$, dropout rates $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and learning rate $\{0.01, 0.05, 0.1, 0.5\}$. Lastly, we combined inputs to FNN with the image latent features learned in classification task, this helps to remove the dependence on the true damage labels provided by an appraiser at inference. The RMSE scores of different models are summarized in Table 4. The regression models were able to achieve comparable RMSE score. However, it can be seen that FNN with image feature does not require information about the true damage labels, which it infers from the image feature. Lastly, Fig. 12 shows a comparison between the natural variance in the dataset as compared to the error made by the models. The mean cost variance of the dataset is higher than the model prediction errors, which means if a customer goes for a repair, the estimate of the appraiser has a variance of approximately ±172. Whereas, the model was able to significantly reduce the variance to approximately ±80.

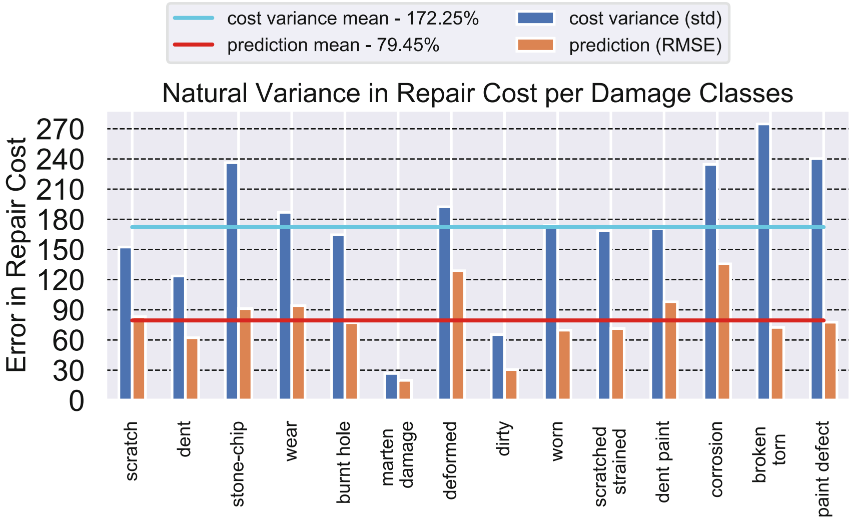


Fig. 12. The comparison of natural variance in the dataset with the prediction made by the model.

5 Conclusion

In this paper, we have presented the challenges encountered when translating the gains made in the field of machine learning to a real-world application and the necessary steps to overcome those challenges. Translating the gain to a propriety dataset requires a data-driven approach to transform the dataset into one that lends itself to machine learning problems. We show how explainable machine learning can be employed to understand the factors causing the machine learning models to under-perform and design a strategy to be applied to similar datasets. This work has also shown a novel application of cost-sensitive loss functions to a new use-case, where widely used cross entropy loss does not capture the important aspects of the task at hand. We experimentally show the gains made by leveraging cross domain knowledge i.e. using bounding boxes to improve classification accuracy. Lastly, we developed a cost regression solution, which leverages latent features from both images and vehicle feature to improve the regression task. We were able to significantly reduce variance in the cost estimation as compare to the manual estimations by appraisers.

Acknowledgment. This work is co-funded by the industry project "Data-driven Mobility Services" of ISMLL and Volkswagen Financial Services (https://www.ismll.uni-hildesheim.de/projekte/dna_en.html).

References

1. Berger, R.: Embracing the caras-a-service model - the European leasing and fleet management market, January 2018. https://www.rolandberger.com/publications/publication_pdf/roland_berger_car_as_a_service_final.pdf
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
3. Chen, T., Guestrin, C.: XGboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 785–794. Association for Computing Machinery (2016)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR 2016, pp. 770–778 (2016)
5. Li, P., Shen, B., Dong, W.: An anti-fraud system for car insurance claim based on visual evidence. arXiv preprint arXiv:1804.11207 (2018)
6. Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H.: Road damage detection and classification using deep neural networks with smartphone images. Comput.-Aided Civ. Infrastr. Eng. **33**(12), 1127–1141 (2018)
7. Nia, K.R., Mori, G.: Building damage assessment using deep learning and ground-level image data, pp. 95–102. IEEE, May 2017
8. Patil, K., Kulkarni, M., Sriraman, A., Karande, S.: Deep learning based car damage classification. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 50–54, December 2017
9. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
10. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626 (2017)
11. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31, pp. 527–538. Curran Associates, Inc. (2018)
12. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2016)