# A System for Automated Vehicle Damage Localization and Severity Estimation Using Deep Learning

Yuntao Ma, Hiva Ghanbari, Tianyuan Huang, Jeremy Irvin, Oliver Brady, Sofian Zalouk, Hao Sheng, Andrew Ng, Ram Rajagopal, *Member, IEEE*, and Mayur Narsude

***Abstract*— Vehicle damage localization and severity estimation is essential to post-accident assessments, with a traditional process taking an average of seven days and requiring substantial work from both customers and dealers. Towards improving this process, we propose an end-to-end system which inputs a set of user-acquired photographs of a vehicle after an accident and outputs a damage assessment report including the set of damaged parts and the type and size of the damage for each part. The system is composed of three deep learning modules: a model to identify whether a vehicle is present in the image, a model to localize the vehicle parts in the image, and a model to localize the damage in the image. We demonstrate the effectiveness of each module by evaluating them on labeled datasets containing images of vehicles after an accident, some collected by the OE (Original Equipment) Insured Fleet and some acquired by users of the OEM (Original Equipment Manufacturer) mobile application. We also describe how the modules fit together with a post-processing step to aggregate outputs between the different modules across multiple user-acquired views of the accident. Our approach demonstrates the potential for an accurate and automated vehicle damage estimation system to support a substantially more efficient vehicle damage assessment process.**

***Index Terms*— Vehicle part localization, vehicle damage estimation, deep learning, computer vision, machine learning.**

## I. Introduction

APPROXIMATELY 6.75 million car accidents occur every year in the U.S. according to the Bureau of Transportation Statistics (BTS). Damage assessment after the accident is a time-intensive process: estimating the collision body repair cost takes seven days on average [22] and includes the customer submitting the First Notice of Loss (FNOL), the adjuster evaluating claim details, then a field appraiser inspecting the unit and preparing the report. Furthermore, this process is subject to inconsistent mistakes due to human error and bias.

There is an opportunity to leverage user-acquired photographs of the damaged vehicle to substantially expedite the damage assessment process. An automated approach to analyze external images of the damaged vehicle to accurately and efficiently identify, localize, and categorize the damaged parts of the vehicle would drastically reduce the amount of time needed for the assessment process. This approach is a key feature of a new program to provide seamless real-time post-accident customer service. Such technology would make the post-accident process more efficient for customers and assessment workers, improving customer satisfaction and providing customers with the opportunity to receive original parts and service repairs by Collision Repair Centers. Moreover, this approach has the potential to provide more fair and uniform inspection and evaluation to customers.

Deep learning powered computer vision technologies have the potential to enable such an automated damage localization approach. Breakthroughs in computer vision have driven several recent significant advancements within the field of AI [24], [26]. These methods have demonstrated immense success across a variety of fields including healthcare, remote sensing, autonomous vehicles, and manufacturing, often achieving performance rivaling human experts [12], [25], [33], [38].

Many previous works have developed deep learning models on images of vehicles for vehicle part and damage identification. Early works developed convolutional neural networks to classify images of vehicles for the presence of a scratch [5] and to classify images of vehicles into seven different types of damage categories including bumper dent, door dent, glass shatter, head-lamp broken, tail-lamp broken, scratch, smashed, and a no damage class [14]. A similar work developed a model to classify the size of the damage into medium, huge, or no damage [3]. Studies have begun to explore the development of multiple deep learning models for vehicle damage localization, with several studies dividing the task into three subtasks: classifying the presence of any damage in the image, classifying where in the image the damage occurred as front, rear, or side, classifying the severity of the damage into minor, moderate, and major [9], [15]. One work adopted a similar strategy for the first two modules but modified the

Manuscript received 14 March 2023; revised 9 October 2023; accepted 9 November 2023. Date of publication 1 December 2023; date of current version 31 May 2024. This work was supported by Ford Motor Company. The Associate Editor for this article was M. Guo. *(Corresponding author: Yuntao Ma.)*

Yuntao Ma is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: yma42@stanford.edu).

Hiva Ghanbari and Mayur Narsude are with the Global Data Insights and Analytics, Ford Motor Company, Dearborn, MI 48124 USA.

Tianyuan Huang is with the Department of Civil Engineering, Stanford University, Stanford, CA 94305 USA.

Jeremy Irvin, Oliver Brady, Sofian Zalouk, and Andrew Ng are with the Department of Computer Science, Stanford University, Stanford, CA 94305 USA.

Hao Sheng was with the Department of Computer Science, Stanford University, Stanford, CA 94305 USA. He is now with Apple, Cupertino, CA 95014 USA.

Ram Rajagopal is with the Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94304 USA.

Digital Object Identifier 10.1109/TITS.2023.3334616

final module to output eight damage severity classes ranging from minor to severe [27]. While these approaches have shown promising results, the models are all framed as independent, single-label classification tasks and therefore they are not able to detect multiple types of damage in different locations on the vehicle. Our approach includes deep learning modules for vehicle part and damage localizations, allowing the system to produce an arbitrary number of damaged parts.

Several studies have additionally developed approaches to localize specific areas of the vehicle that are damaged, with one notable study proposing the three subtasks approach with an additional final instance segmentation module to outline the damaged areas in the image [20] and a recent study adopting a similar approach but training the final instance segmentation model to classify the type of damage into scratch, dent, shatter, and dislocation [1]. One study framed the problem as a binary semantic segmentation task, where the objective is to identify which pixels of the image are damaged [23]. Studies have also begun to explore the development of vehicle part identification models, including a standalone localization approach into 18 vehicle part categories [13], a classification approach of specific parts into three categories (rear bumper, car wheel, front bumper) combined with damage classification [2], and an approach to localize and crop the image to the vehicle and classify whether the vehicle is damaged [16]. Across all of these studies, the choice of how to frame the vehicle damage localization approach has substantial implications on model accuracy and usability. Furthermore, none have proposed a system designed around the machine learning model to make it usable within and beneficial to the accident assessment workflow.

In this work, we design a system comprised of multiple deep learning models for automated vehicle damage localization and severity estimation in user-acquired photographs of vehicles. Our contributions are as follows:

1) We design a three-module approach to vehicle damage localization and severity estimation and train it using novel datasets capturing user-acquired photographs that we carefully curate labels for using crowd-sourcing.
2) We run several experiments testing the performance of the different modules, including investigating the effect of joint damage and vehicle part training compared to independent training as well as the impact of training dataset size on performance.
3) We describe a novel post-processing procedure to aggregate and process the outputs of the modules, including automatically identifying the perspective of user-acquired photographs in order to estimate the size of the damage, synthesizing the damage predictions across multiple views, and determining the confidence score for integration in a human-in-the-loop system.

In Section II we describe the justification for the three-module design and how they work together (Section II-A), the data used to develop each of the modules (Section II-B), the details of the deep learning models (Section II-C), and how the deep learning model outputs are post-processed and synthesized across multiple use-acquired images to generate a damage report as part of a human-in-the-loop workflow (Section II-D). We quantitatively and qualitatively assess each of the modules in Section III and finally discuss the implications, limitations, and interesting future directions of our work in Section IV.

## II. Methodology

### A. Modules

The goal of the system is to, given a user-acquired photograph, produce a report detailing which parts of the vehicle are damaged along with the type and size of the damage for each damaged part. This end-to-end system must be able to produce accurate results on general user-acquired photographs, potentially not capturing a vehicle, and identify which parts are damaged and how they are damaged. To address, this, we design the system to be composed of three primary modules, namely:

1) *Vehicle Identification*: A binary classification model to identify whether a vehicle is in the image,
2) *Vehicle Part Localization*: A vehicle part semantic segmentation model to localize where different vehicle parts occur in the image,
3) *Vehicle Damage Localization*: A damage semantic segmentation model to localize where various damages occur on the vehicle in the image.

To handle images without vehicles, the first part of the system performs vehicle identification to filter out images without vehicles, which several other works have adopted as well [1], [9], [15], [20], [27]. We believe that the skill required for models to perform vehicle identification is substantially different from vehicle part and damage localization, and therefore use a separate module for this task.

To generate a list of damaged parts with their severity, we argue that a natural and flexible approach is for the system to separately localize the parts and localize the damages, then overlay these predictions to produce the list of damaged parts. This framing allows the model to produce an arbitrary number of damaged parts which are simple to obtain by combining the predicted masks between the two localization modules. An alternative framing would be to use a single end-to-end module that can output damage parts and types together, for example ‘scratched door’ or ‘dented hood’. However, coupling the part and damage type prevents the system from sharing information between parts with the same type of damage, for example ‘scratched door’ and ‘scratched hood’ would be separate classes even though the damage type is the same. Decoupling these predictions allows the model to share information and enables more flexibility. Fully decoupling them still may not be ideal, however, since the vehicle part localization and damage localization tasks may benefit from shared data and features. To investigate this, we explore the development of joint deep learning models that share parameters between the two tasks (see the “Independent versus Joint Localization Models” subsection of Section II-C). Finally, it is worth noting the modules cannot be used on their own (e.g. a vehicle damage localization module alone) as both the parts and the damage types are required in the report.

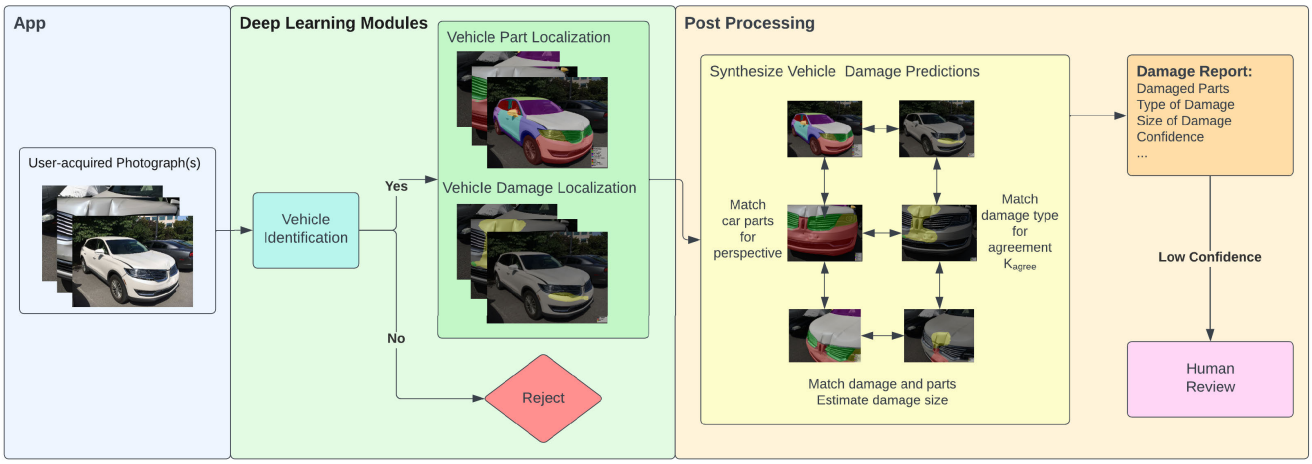



Fig. 1. Full end-to-end damage assessment system using user-acquired photographs. A user acquires photographs of their vehicles which are fed into a vehicle identification module to retain images of the vehicle. Then a vehicle part and damage localization model are used to identify the locations of the vehicle parts and damage respectively. Finally, we conduct post-processing that we first uses the car part localization between different views to estimate the perspective of each view then aligns the car part and damage type localizations from the same image to assign damage to parts and estimate size of the damaged area based on the number of car parts. We compare the damage type localization between views to establish an agreement factor between predictions $K_{agree}$.

Under this three module design, we employ the following procedure to produce a damage report for each image: the image is first input to the vehicle identification module to filter out any images which do not contain a vehicle. Then the image is fed into the vehicle part and damage localization modules whose outputs are overlaid to produce a list of damaged parts, each classified with the type and severity of damage as well as a score indicating the confidence of the model in that prediction. Any predictions made with low confidence are sent to a human for manual review. The full system is shown in Figure 1.

### *B. Data*

To develop and validate the vehicle presence classification model, we used images collected by the OE Insured Fleet and user-uploaded pictures obtained from the OEM mobile application containing Sedans, SUVs and Trucks. Customer consent was obtained to use the data for research purposes without revealing any Personal Identifiable Information (PII). OE Insured Fleet images tend to be higher quality whereas OEM images tend to be lower quality as they are images captured by actual users in practice. Binary classification labels were obtained by manually classifying each image as containing a vehicle or not. The total dataset for the vehicle classification model has 17,215 OE Insured Fleet images with a vehicle and 7,828 OE Insured Fleet images with no vehicle, as well as 402 OEM images with a vehicle and 307 images with no vehicle. We split the dataset into a training set (90% of the OE Insured Fleet images) to learn vehicle classification model parameters, a validation set (10% of the OE Insured Fleet images) to compare the models, and a test set (100% of the OEM images) to evaluate the best model on previously unseen data similar to data captured in practice.

To develop and validate the vehicle part and damage localization models, we also conducted experiments on the OE Insured Fleet and OEM data. To obtain segmentation labels for the images, we used a crowd-sourcing platform called MolarData. To ensure the quality and consistency of the image labels, labelers were required to carefully review both of the following resources, including (1) a data annotation training document that provided detailed guidelines for annotations, ensuring that labelers were well-informed about our expectations and (2) a set of 20 pre-labeled images which served as examples to train the labelers and increase their accuracy and consistency. Examples of OE Insured Fleet and OEM images and corresponding segmentation masks for vehicle part and damage localization are shown in Figure 2. After conducting several experiments investigating the effect of label merging strategy on model performance, we designed the final annotation taxonomy for vehicle parts to have 13 categories (plus background) and the taxonomy for damage to have 3 damage types (plus no damage). The full taxonomy with counts per split is shown in Table I. After reviewing the dataset, we found that some of the images, especially the user-uploaded ones, were acquired from poor perspectives making it difficult to identify the vehicle parts and damage. For example, images that focus on the vehicle interior or images that fail to capture the entire damaged part. After filtering the data with poor perspectives, the resulting dataset has 6,717 OE Insured Fleet images and 732 OEM images. OE Insured Fleet labeled data was split randomly into a training set (5374 samples, 80%), a validation set (672 samples, 10%), and a test set (671 samples, 10%). We used all OEM data as additional test set. Since multiple images could come from the same accident case in OE Insured Fleet and OEM, we ensured that images from the same accident case are assigned to the same split to prevent information leakage between splits.

### *C. Models*

*1) Vehicle Presence Classification:* In order to identify if an image contains a vehicle, we experimented with several convolutional neural network models. Specifically, we explored various architectures including VGG16, VGG19, ResNet50, InceptionResNetv2, Inceptionv3, MobileNet and MobileNetv2

TABLE I

COUNTS AND PROPORTIONS OF EACH CATEGORY IN THE DATASETS USED FOR VEHICLE PART AND VEHICLE DAMAGE LOCALIZATION. THE TRAINING, VALIDATION, AND TEST SETS INCLUDE IMAGES COLLECTED BY THE OE INSURED FLEET WHEREAS THE OEM TEST SET INCLUDES IMAGES ACQUIRED BY USERS OF THE OEM MOBILE APPLICATION. SEE THE APPENDIX FOR DEFINITIONS OF THE VEHICLE PART AND DAMAGE CATEGORIES

| Task | Category | Train (%) | Valid (%) | Test (%) | OEM Test (%) | Total (%) |
|---|---|---|---|---|---|---|
| Vehicle Part Localization | Bumper | 4849 (90.2) | 609 (90.6) | 597 (89.0) | 562 (76.8) | 6617 (88.8) |
| | Grille | 2104 (39.2) | 258 (38.4) | 258 (38.5) | 208 (28.4) | 2828 (38.0) |
| | Lamps | 4726 (87.9) | 579 (86.2) | 592 (88.2) | 501 (68.4) | 6398 (85.9) |
| | Fender | 4174 (77.7) | 535 (79.6) | 525 (78.2) | 356 (48.6) | 5590 (75.0) |
| | Windshield | 2291 (42.6) | 272 (40.5) | 279 (41.6) | 250 (34.2) | 3092 (41.5) |
| | Door Shell | 4527 (84.2) | 563 (83.8) | 552 (82.3) | 394 (53.8) | 6036 (81.0) |
| | Hood | 2689 (50.0) | 334 (49.7) | 339 (50.5) | 340 (46.4) | 3702 (49.7) |
| | Wheel | 4857 (90.4) | 610 (90.8) | 597 (89.0) | 505 (69.0) | 6569 (88.2) |
| | Window | 4334 (80.6) | 551 (82.0) | 526 (78.4) | 343 (46.9) | 5754 (77.2) |
| | Roof | 3301 (61.4) | 409 (60.9) | 389 (58.0) | 245 (33.5) | 4344 (58.3) |
| | Mirror | 4017 (74.7) | 509 (75.7) | 497 (74.1) | 276 (37.7) | 5299 (71.1) |
| | Panel | 4856 (90.4) | 601 (89.4) | 601 (89.6) | 514 (70.2) | 6572 (88.2) |
| | Trunk | 2455 (45.7) | 310 (46.1) | 303 (45.2) | 236 (32.2) | 3304 (44.4) |
| Damage Localization | Surface Damage | 1257 (23.4) | 162 (24.1) | 152 (22.7) | 313 (42.8) | 1884 (25.3) |
| | Deformity | 1309 (24.4) | 172 (25.6) | 143 (21.3) | 384 (52.5) | 2008 (27.0) |
| | Body Damage | 933 (17.4) | 110 (16.4) | 105 (15.6) | 347 (47.4) | 1495 (20.1) |
| Total Number of Images | | 5374 | 672 | 671 | 732 | 7449 |

[10], [11], [28], [29], [31], [32]. We replaced the final linear layer with a layer that outputs a single score indicating the predicted likelihood that the image contains a vehicle. We initialized the network with weights learned from training the model on ImageNet [8]. We used an unweighted binary cross entropy loss function, an RMSprop optimizer, a learning rate of 0.0001, and a batch size of 16. We trained for 200 epochs and saved the checkpoint which achieves the highest accuracy on the validation set.

The best vehicle presence classification model was a MobileNet, achieving an accuracy of 0.989 on the OE Insured Fleet validation set and 0.910 on the OEM test set.

*2) Vehicle Part Localization:* To localize the vehicle parts in each image, we utilized an encoder-decoder network segmentation architecture. Specifically, we used DeepLabV3+ [6] as the segmentation architecture and EfficientNet-b5 as the convolutional backbone encoder, which together demonstrated the highest performance in preliminary experiments compared to other segmentation architectures including UNet++ [37] and PSPNet [36] as well as backbone architectures including ResNet [10] and and ResNeXt [34] (see Appendix). We replaced the segmentation head of each architecture to output the correct number of classes.

To increase the representativeness of the dataset and improve model robustness to variations in imagery including perspectives and lighting conditions, we performed extensive data augmentations using the albumentations library [4]. For each input image, we applied the following augmentations during training including random horizontal flips, random shifts between $-0.1$ and 0.1 and scales between $-0.5$ and 0.5 (values outside the border were filled with 0), random crops to $512 \times 512$ to account for variable input image sizes, additive Gaussian noise with 0 mean and standard deviation between 2.55 and 12.75 (applied with a 0.2 probability) and random four point perspective transforms with standard deviation of normal distributions between 0.05 and 0.1 (applied with a 0.6 probability). We additionally applied the following augmentations with a 0.9 probability of the set being applied to the input image:

- One of contrast limited adaptive histogram equalization with an upper threshold clip value of 4 and tile grid size of $8 \times 8$, random brightness with a factor between $-0.2$ and 0.2, or random gamma transform with a gamma limit between 80 and 120,
- One of sharpening overlay with an alpha value between 0.2 and 0.5 and lightness between 0.5 and 1.0, blurring with a random kernel size between 3 and 7, or motion blurring with random kernel size between 3 and 7,
- One of random contrast with a factor between $-0.2$ and 0.2, or random hue saturation value shift with a hue shift limit between $-20$ and 20, saturation shift limit between $-30$ and 30, and value shift limit between $-20$ and 20.

If a set is applied, a single one of the transforms in the set is chosen with equal probability.

We additionally experimented with three common loss functions used for semantic segmentation including cross entropy loss [7], dice loss [30], and focal loss [17], and found that dice loss produced the best results (see Appendix). For all models, we initialized the backbone encoder with weights from a network trained on ImageNet [8]. We used an Adam optimizer with a starting learning rate of 0.0001 and linear learning rate warmup over first 10 epochs, and decayed the learning rate with the one cycle cosine annealing for the rest of the training process. All input images were augmented during training using the approach described in Section II-C. We used a batch size 4 and trained models for 50 epochs, and saved the checkpoint which achieved the highest total mIoU on the validation set. All models were trained using a single NVIDIA RTX A4000 GPU.

After the model is trained, new images are input by first resizing the longer side to 512 pixels while keeping the aspect ratio of the input image the same. The model then operates on this resized image to output a mask of the same size as the resized image, where each pixel is associated with a set of


Fig. 2. Examples of OE Insured Fleet and OEM images and their corresponding vehicle part and damage type labels. The leftmost image in each row is the original image, the middle has the vehicle part labels overlaid, and the rightmost has the damaged regions overlaid.

scores indicating the likelihood the pixel corresponds to the class. The class assigned the maximum score by the model is used as the prediction for each pixel.

*3) Vehicle Damage Localization:* We used the same model architectures, data augmentations, training procedure, and inference procedure for the vehicle damage localization model.

*4) Independent Versus Joint Localization Models:* We hypothesized that the vehicle damage and vehicle part localization tasks may benefit from sharing features. Intuitively, the vehicle damage type will often be unique to specific vehicle parts. For example, a broken window could only exist on a windshield or vehicle windows, and paint chips will only exist on panels. To this end, we investigated the use of a shared encoder and decoder between the vehicle part and damage localization segmentation models. We use a separate segmentation heads to compute the per-pixel scores for the vehicle parts and vehicle damages which we then feed through two separate softmax functions to derive the per-pixel vehicle part and vehicle damage probabilities $\hat{p}_{\text{vehicle\_parts}}$ and $\hat{p}_{\text{damage}}$ respectively. The loss was the sum of the vehicle part localization and vehicle damage localization losses:

$$\ell_{\text{total}} = (1-\lambda)\ell_{\text{dice}}(y_{\text{vehicle\_parts}}, \hat{p}_{\text{vehicle\_parts}}) + \lambda\ell_{\text{dice}}(y_{\text{damage}}, \hat{p}_{\text{damage}})$$

We weight both losses equally by setting $\lambda = 0.5$.

### D. Post-Processing

This section describes the post-processing of the model predictions to estimate the size of the damage and synthesize the vehicle damage predictions across multiple user-acquired photographs which are part of a single case.

*1) Damaged Part Estimation:* We combined the outputs of the modules to determine the damaged parts as follows. First,

we computed the per-pixel argmax of the vehicle part predicted probabilities $\hat{p}_{\text{vehicle\_parts}}$ and the damage type predicted probabilities $\hat{p}_{\text{damage}}$ to obtain the predicted classifications $\hat{y}_{\text{vehicle\_parts}}$ and $\hat{y}_{\text{damage}}$ respectively. Then we computed their intersection $\hat{y}_{\text{damaged\_part}}$ where each pixel represents a unique combination of vehicle part and damage type:

$$\hat{y}_{\text{damaged\_part}}(i, j) = (\hat{y}_{\text{vehicle\_parts}}(i, j), \hat{y}_{\text{damage}}(i, j))$$

Then we can define each unique damaged part $\hat{D}_{\text{k}}$ as the set of connected neighboring pixels that have the same value pair in $\hat{y}_{\text{damaged\_part}}$.

*2) Damage Size Estimation:* The size of the damage is an important factor to determine the severity and eventually the cost of repair. However, since a user could take a picture from any perspective and from any distance, this makes estimating the size of damage difficult. For example, the size of the damage in a close-up image with many predicted damage pixels could be small compared to a zoomed-out image with less predicted damage pixels.

To circumvent this, we leveraged vehicle part localization to be able to obtain a damage size estimation. We found that the number of vehicle parts predicted in the image is a good surrogate for the distance from where the image was taken to the vehicle. When there are more predicted vehicle parts present in the image, we can confidently say it is a full-body shot of the vehicle. When there are less predicted vehicle parts present in the image, we can assume it is a close-up view of the vehicle. Once determining the perspective of the vehicle, the number of predicted pixels can be used to estimate the size of the damage.

*3) Synthesizing Vehicle Damage Predictions:* Users often submit multiple images in a single accident case capturing the vehicle and damaged part(s) from multiple perspectives. We can leverage this fact to validate and synthesize predictions on images within the same case. For example, if the two images contain the same predicted vehicle parts, they likely depict similar perspectives of the vehicle. If one image contains parts which are a subset of the parts in another image, we conclude the first image is a close-up of the second image. Then, high agreement between differing perspectives indicates higher confidence in the model predictions and low agreement indicates lower confidence. A low confident prediction will be passed to a human for manual review.

*4) Confidence Estimation and Human-in-the-Loop Workflow:* We combined both statistical and heuristic methods to assess the confidence level of our damage predictions. When users submitted a single view, the base confidence score $\hat{C}_{\text{base},k}$ for each damaged part $\hat{D}_{\text{k}}$ was calculated by averaging the per-pixel damage type probability $\hat{p}_{\text{damage,n}}$ within $\hat{D}_{\text{k}}$

$$\hat{C}_{\text{base},k} = \frac{1}{N} \sum_{n \in \hat{D}_{\text{k}}} \hat{p}_{\text{damage},n}$$

When users submit multiple images, these base confidence scores are further refined using the synthesized vehicle damage predictions. A high level of agreement between different images of the same case is indicative of higher confidence, whereas low agreement suggests lower confidence. We defined agreement factor $K_{\text{agree}}$, which is the number of agreed images over all user submitted images. The refined confidence $\hat{C}_{\text{refined}}$ is calculated as the product of the base confidence $\hat{C}_{\text{base}}$ and an agreement factor $K_{\text{agree}}$.

$$\hat{C}_{\text{refined}} = \hat{C}_{\text{base}} K_{\text{agree}}$$

Predictions that fall below a predetermined confidence threshold are flagged for human review. This threshold is empirically set by manually inspecting predictions on a small test set collected prospectively in actual operations and is reset every model version.

### E. Evaluation

We primarily evaluate the vehicle part and damage localization models using intersection over union (IoU), otherwise known as Jaccard index, which is a commonly used metric to assess the quality of segmentation predictions. The metric is formally defined below as follows for a ground truth segmentation mask $y$ and predicted segmentation mask $\hat{y}$:

$$IoU(y, \hat{y}) = \frac{\|y \bigcap \hat{y}\|}{\|y \bigcup \hat{y}\|} \tag{1}$$

To assess for statistical differences between the independent and joint localization models, we used the nonparametric bootstrap. Specifically, we sampled 5,000 bootstrap replicates from the validation (test) set and computed the difference in mIoU between the models on each replicate. If the interval was strictly positive (did not include 0), there was evidence that the model was superior.

## III. Results

### A. Independent Versus Joint Localization Models

The independent vehicle part localization model was statistically significantly better than the joint model across the vehicle part classes and the joint model was statistically significantly better than the independent damage type localization model across the damage classes on the validation set (Figure 3). Specifically, for vehicle part localization the bootstrapped difference between the independent and joint model was 0.0168 (95%CI, 0.0004 - 0.0231) and for damage localization was 0.0062 (95%CI, 0.0013 - 0.0181). The independent vehicle part localization model outperformed the joint one by a small margin on 13 out of the 14 classes, whereas the joint model outperformed the damage localization model by a small margin on 3 out of the 4 classes, with a very small drop in performance on the No Damage class. This suggests that damage labels may not be useful for identifying vehicle parts but vehicle part labels may be useful for differentiating types of damage.

### B. Effect of Training Set Size

Collecting and labeling sufficient data for variety of vehicles and damage types is time-consuming and expensive. In order to measure how much labeled data is necessary for achieving the model's level of performance and estimate how much of an impact including more labeled data could make on

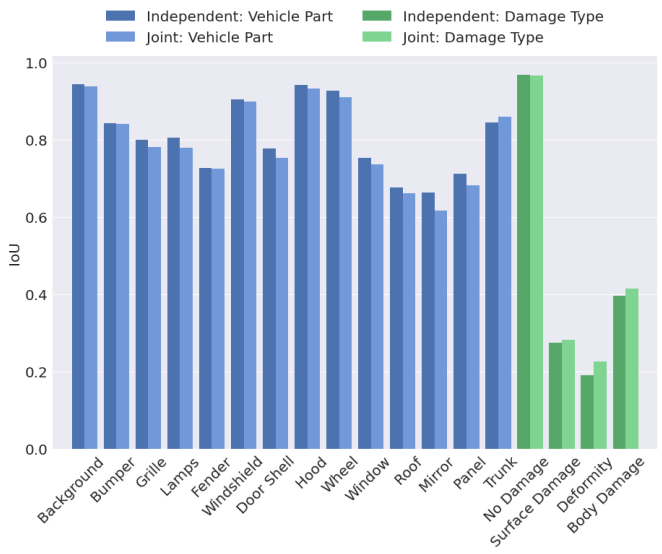



Fig. 3. Independent versus joint model per-class IoU on the validation set for vehicle part and damage type localization. The joint model shares both the encoder and decoder with different segmentation heads whereas the independent models share no parameters.

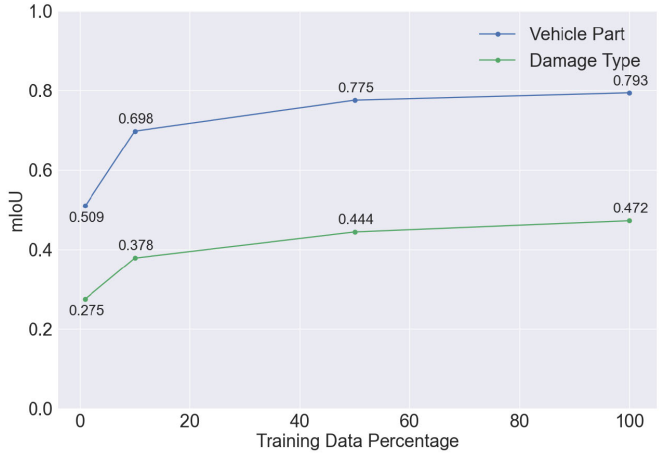



Fig. 4. Validation set performance in terms of mean IoU for vehicle part and damage localization when training with 1%, 10%, 50%, and 100% of the training data.

model performance, we assessed model performance using 1% (54 examples), 10% (536 examples), 50% (2,687 examples) of the training set and compared it to performance with 100% of the set (5,374 examples).

Performance substantially improved from 1% to 10% and from 10% to 100% of the training data for both vehicle part and vehicle damage localization (Figure 4). The vehicle part localization performance increases from 0.509 to 0.698 to 0.793 mean IoU across the classes (mIoU) for 1%, 10%, and 100% respectively whereas the damage types localization performance increases from 0.275 to 0.378 to 0.472 mIoU respectively. Increasing the dataset size by another order of magnitude may continue to lead to substantial performance improvements. Performance at 100% of the data increases slightly compared to 50%, from 0.775 to 0.793 and from 0.444 to 0.472 for vehicle part and damage localization, respectively.

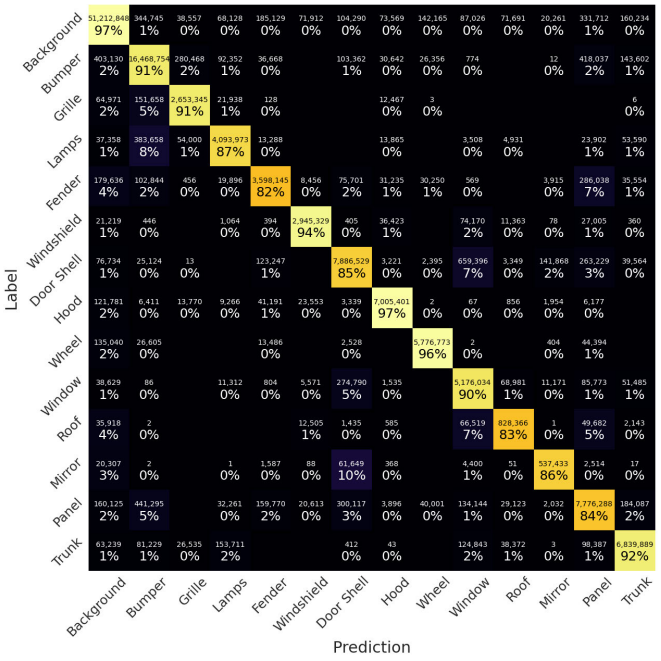



Fig. 5. Confusion matrix for vehicle part localization on the validation set. The value in each cell indicates the number of pixels predicted and the percentage of all labeled pixels for that class. Empty cells indicate no prediction is made for that class.

### C. Best Model Performance on the Validation Set

We used the best performing model on the validation set for further evaluation, specifically the independent vehicle part model (0.808 mIoU) for the vehicle part localization task and the joint model (0.472 mIoU) for the vehicle damage localization task. We investigated model errors on the validation set to identify common patterns of model mistakes. The confusion matrix for the vehicle car localization task (Figure 5) indicates parts that are similar often misclassified together. For example, 10% of the Mirror class pixels were misclassified as Door Shell, 7% of the Door Shell class pixels were misclassified as Window, and 5% of Window class pixels were misclassified as Door Shell, and each of these vehicle parts all contain glass components. We also saw minor misclassifications occur for adjacent vehicle parts. For example, 5% and 1% of the Grille pixels were misclassified as Bumper and Lamps respectively, and Bumper and Lamps are physically adjacent to the Grille. This could indicate boundary errors between these classes. Furthermore, approximately 32.8% of the misclassified pixels were incorrectly classified because of adjacency to other parts and approximately 65.9% of the non-background misclassified pixels were misclassified due to similarity to other parts.

The confusion matrix of the damage types localization (Figure 6) shows the majority of the model mistakes were misclassifying the damage types as no damage and no damage as being damaged (>2 million pixels in total for each). Generally the model gets the type of damage correct when correctly predicting damage, with the highest amount of incorrectly predicted being Deformity when the damage was Body Damage. Of the damage cases the model misclassified as no damage, 30%, 53%, and 57% of Body Damage, Surface Damage, and Deformity pixels were misclassified as no damage respectively.

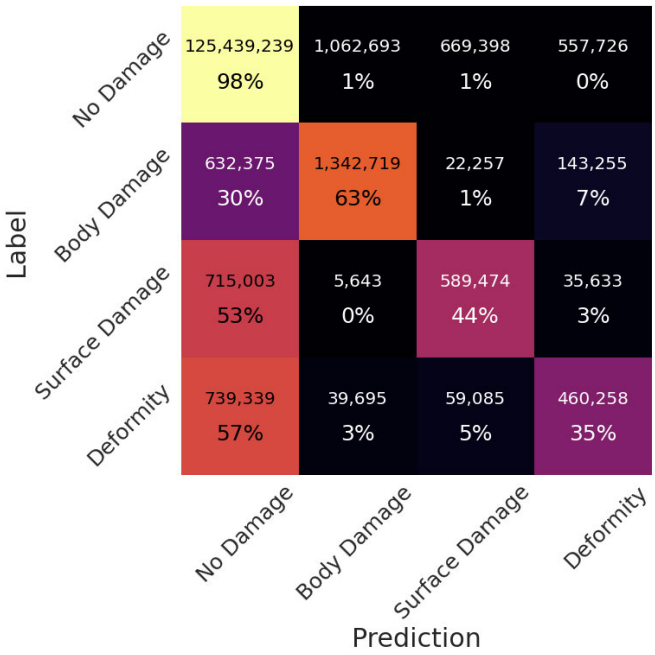



Fig. 6. Validation set confusion matrix for damage types localization, value indicates number of pixel predicted, percentage normalized row-wise.

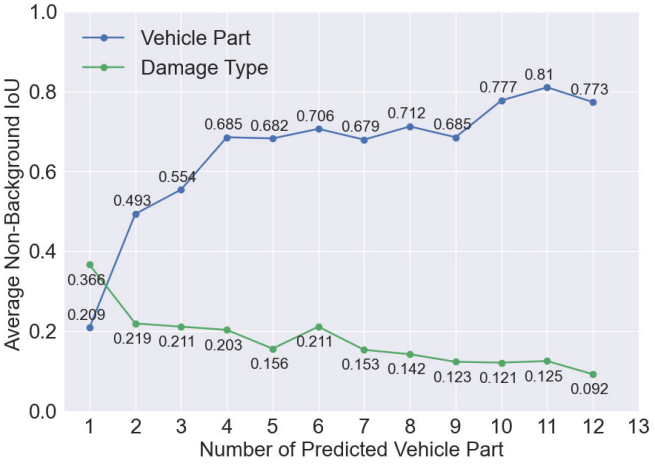



Fig. 7. Average non-background class IoU by predicted number of vehicle parts in each image within the validation set.

### D. Post Processing Assessment on the Validation Set

In our post processing, we used predicted vehicle parts as a surrogate to the distance from where the image was taken to the vehicle. We used the best performing model on the validation set for further evaluation on accuracy of predicting number of vehicle parts. We found that 411 out of 672 (61.2%) predictions on the validation set had the correct number of vehicle parts, 615 (91.5%) predictions were within $\pm 1$ of the correct number of vehicle parts, and all prediction were within $\pm 3$ of the true number of parts.

We also examined the performance of vehicle parts and damage type localization at different numbers of predicted vehicle parts (Figure 7). Vehicle part localization performed better with a zoomed-out, full-body shot of the vehicle compared to close-up images. The average non-background vehicle parts IoU increased vastly from 0.209 to 0.493 to 0.674 with 1, 2, and 4 predicted number of vehicle parts. The performance stayed relatively similar from 4 to 9 predicted vehicle parts, and there was a small performance increase from 0.685 to 0.810 with 9 and 11 predicted number of vehicle parts. On the contrast, damage type localization performed better with close-up images compared to zoomed-out full-body images. The average non-background damage types IoU increased steadily from 12 to 2 predicted vehicle parts, achieving 0.092 and 0.219 average IoU respectively. The performance increased substantially from 2 to 1 predicted vehicle parts, achieving 0.219 and 0.366 IoU respectively.

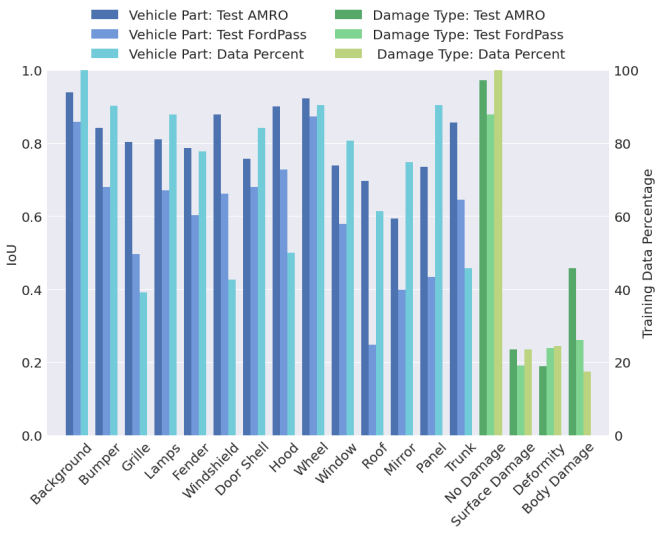



Fig. 8. Per-class IoU scores of the best performing model for vehicle part and damage localization on the OE Insured Fleet and OEM test sets and corresponding class prevalence in the training split.

### E. Best Model Performance on the Test Sets

The model which performed best on the validation set achieves very similar performance on the OE Insured Fleet test set (Figure 8). The overall mIoU on the OE Insured Fleet test set for the vehicle part localization task was 0.804 and for the damage localization task was 0.463.

Vehicle part localization performance on the OE Insured Fleet test set was higher than 0.7 IoU across 12 out of the 14 classes (Figure 8). The highest performance was achieved on the background, wheel, hood, and windshield classes, achieving 0.940, 0.922, 0.900, and 0.878 IoUs respectively. These are all large and clearly differentiable parts on the vehicle. The lowest performance was achieved on the mirror and roof classes with IoUs of 0.594 and 0.696 respectively. These parts are both small and can be difficult to differentiate from other parts.

Vehicle damage localization performance on the OE Insured Fleet test set was close to 0.2 IoU or above for all damage classes (Figure 8). The highest performing class was no damage (0.973 IoU) which constitutes the majority of the pixels in the image, followed by body damage (0.457 IoU), surface damage (0.234 IoU), and deformity (0.189 IoU).

Performance on vehicle part and damage type classes with higher number of image examples in the dataset is generally higher than those with fewer examples (Figure 8). Notable exceptions include grille, windshield, and hood on which the model achieves high performance (0.803, 0.878, and 0.901 IoU respectively) with less than half of the images in the dataset having labels for these classes. These vehicle parts tend to be large and easy to identify compared to the other parts.

There was a substantial drop in performance from the OE Insured Fleet test set to the OEM test set (Figure 8). The overall IoU on the OEM test set for the vehicle part localization task was 0.611 and for the damage localization task was 0.392. This suggests there is a distribution shift from the OE Insured Fleet data to the OEM data. However, this phenomenon is more pronounced for vehicle part localization which exhibits large drops in performance across all classes compared to damage localization where performance is less affected. We hypothesize that although OE Insured Fleet captured images and user-acquired images may lead to substantial differences in appearance and heterogeneity of vehicle parts, damage appearances may be more consistent between the images. This could have an important implication in practice, where when a new vehicle model is released, the damage localization model could be applied without much additional tuning.

## IV. Discussion and Conclusion

In this work, we proposed a framework using computer vision to systematically localize vehicle damage and estimate the damage severity. We have - for the first time - integrated multiple deep learning modules together within a human-in-the-loop system for automated vehicle damage localization and severity estimation in user-acquired photographs of vehicles. By evaluating our results on both corporate-owned OE Insured Fleet data and user-acquired OEM data, we demonstrate the potential of our proposed approach to further support a more automated and accurate vehicle damage assessment platform.

While our data-driven method provides an end-to-end solution to the localization of vehicle damage and damage severity estimation, it is still subject to the following limitations. First, due to the nature of our proposed framework and the dataset we use, our model is only able to detect visual cues on the vehicle exterior, so damage on parts inside the vehicle such as the engine and transmission cannot be localized. Second, the approach does not estimate repair cost directly, but it does provide a list of damaged parts. The make, model, and year of the vehicle could be combined with the list of damaged parts as a post-processing step to estimate the repair costs post-accident based on the price of the specific parts that need to be replaced. Future work should investigate the development of this integration as well as incorporating this information into the damage localization modules themselves which may help further improve performance. Third, since our approach classifies the most common vehicle damage into three damage categories, it is still a relatively coarse damage estimation compared with the traditional damage assessment approaches. This framing captures the majority of damage cases and costs unlike the majority of prior work, but future work should explore methods to provide a more granular estimation to allow for more accurate repair cost estimation. Fourth, the current system cannot automatically filter out images capturing poor perspectives of vehicles, which users may submit. As part of future work, we plan to automate the poor perspective filtering by adding this capability to the vehicle identification module.

There are several additional interesting directions that can be explored in future work. First, creating concrete guidelines for how users obtain photographs of their vehicle has the potential substantially improve localization performance. For example, users could be advised to take a few images with the full vehicle exterior visible as well as a few images zoomed in on the damage in order to get a more complete assessment of the extent of the damage and its impact on different areas of the vehicle. Furthermore, integrating perspective information into a modern approach like NeRF [18] could allow for even better damage synthesis between images capturing different views. Second, our approach relies on strongly labeled data which can be expensive to collect. Recent advances in self- and semi-supervised learning have the potential to reduce the amount of labels that are needed [21], [35]. Third, it may be difficult for the model to generalize to new models of vehicles released in the future. Beyond collecting new labels for these images, there may be potential to generate synthetic images to get labeled data for free by using simulation-based approaches [19] as well as recent advances in image-conditioned generative modeling to add synthetic damage to images of vehicles [26]. Fourth, although we tried to improve model robustness through extensive data augmentation, it is worthwhile for future work to investigate the resiliency of the model to other input perturbations, for example occlusions or adversarial attacks.

## Appendix

### A. Taxonomy and Label Merging Strategy

In our initial experiments to develop and validate the vehicle part and damage localization models, we explored the use of a fine-grained taxonomy. For example, vehicle parts were separated by their locations (front lamps vs. rare lamps). We designed the original annotation taxonomy for vehicle parts to have 22 categories and the taxonomy for damage to have 7 damage types. The full taxonomy with counts per split is shown in Table II.

In the fine-grained vehicle part localization taxonomy, we identified that many of the classes were semantically similar. For example, front and rare door shell are similar in appearance and often misclassified by the model. In the fine-grained damage type localization taxonomy, four out of seven classes were present in less than 10% of all training examples, namely corrosion (43 examples, 0.8%), paint chips (146 examples, 2.7%), missing (374 examples 7.0%), and crack (405 examples, 7.5%).

To address these issues, we explored different approaches to merge classes together when designing the task taxonomy. For vehicle parts localization, we merged semantically similar classes together, for example by grouping Front Lamps, Fog

TABLE II
COUNTS AND PROPORTIONS OF EACH CATEGORY IN THE DATASETS USED FOR VEHICLE PART AND VEHICLE DAMAGE LOCALIZATION UNDER THE FULL UNMERGED TAXONOMY

| Task | Category | Train (%) | Valid (%) | Test (%) | OEM Test (%) | Total |
|---|---|---|---|---|---|---|
| Vehicle Part Localization | Front Bumper | 2561 (47.7) | 314 (46.7) | 307 (45.8) | 332 (45.4) | 3514 (47.2) |
| | Grille | 2104 (39.2) | 258 (38.4) | 258 (38.5) | 208 (28.4) | 2828 (38.0) |
| | Front Lamps | 2431 (45.2) | 292 (43.5) | 303 (45.2) | 302 (41.3) | 3328 (44.7) |
| | Fog Lamps | 1870 (34.8) | 237 (35.3) | 238 (35.5) | 174 (23.8) | 2519 (33.8) |
| | Front Fender | 4174 (77.7) | 535 (79.6) | 525 (78.2) | 356 (48.6) | 5590 (75.0) |
| | Windshield | 2291 (42.6) | 272 (40.5) | 279 (41.6) | 250 (34.2) | 3092 (41.5) |
| | Front Door Shell | 4337 (80.7) | 538 (80.1) | 534 (79.6) | 346 (47.3) | 5755 (77.3) |
| | Hood | 2689 (50.0) | 334 (49.7) | 339 (50.5) | 340 (46.4) | 3702 (49.7) |
| | Wheel | 4857 (90.4) | 610 (90.8) | 597 (89.0) | 505 (69.0) | 6569 (88.2) |
| | Door Glass | 4053 (75.4) | 513 (76.3) | 487 (72.6) | 280 (38.3) | 5333 (71.6) |
| | Quarter Glass | 2576 (47.9) | 329 (49.0) | 315 (46.9) | 146 (19.9) | 3366 (45.2) |
| | Roof | 3301 (61.4) | 409 (60.9) | 389 (58.0) | 245 (33.5) | 4344 (58.3) |
| | Mirror | 4017 (74.7) | 509 (75.7) | 497 (74.1) | 276 (37.7) | 5299 (71.1) |
| | Windshield Pillar | 3972 (73.9) | 501 (74.6) | 484 (72.1) | 299 (40.8) | 5256 (70.6) |
| | Rocker Panel | 3244 (60.4) | 425 (63.2) | 419 (62.4) | 193 (26.4) | 4281 (57.5) |
| | Back Window | 2240 (41.7) | 298 (44.3) | 283 (42.2) | 149 (20.4) | 2970 (39.9) |
| | Rear Door Shell | 3893 (72.4) | 475 (70.7) | 479 (71.4) | 263 (35.9) | 5110 (68.6) |
| | Rear Lamps | 2698 (50.2) | 340 (50.6) | 333 (49.6) | 215 (29.4) | 3586 (48.1) |
| | Rear Bumper | 2528 (47.0) | 324 (48.2) | 309 (46.1) | 261 (35.7) | 3422 (45.9) |
| | Rear Quarter Panel | 4109 (76.5) | 515 (76.6) | 512 (76.3) | 290 (39.6) | 5426 (72.8) |
| | Lower Rear Cover | 3732 (69.4) | 452 (67.3) | 460 (68.6) | 191 (26.1) | 4835 (64.9) |
| | Trunk | 2455 (45.7) | 310 (46.1) | 303 (45.2) | 236 (32.2) | 3304 (44.4) |
| Damage Localization | Missing | 374 (7.0) | 45 (6.7) | 45 (6.7) | 116 (15.8) | 580 (7.8) |
| | Scratch | 1157 (21.5) | 149 (22.2) | 144 (21.5) | 298 (40.7) | 1748 (23.5) |
| | Dent | 1022 (19.0) | 125 (18.6) | 111 (16.5) | 290 (39.6) | 1548 (20.8) |
| | Corrosion | 43 (0.8) | 1 (0.1) | 4 (0.6) | 21 (2.9) | 69 (0.9) |
| | Crack | 405 (7.5) | 61 (9.1) | 40 (6.0) | 164 (22.4) | 670 (9.0) |
| | Paint Chip | 146 (2.7) | 19 (2.8) | 17 (2.5) | 31 (4.2) | 213 (2.9) |
| | Broken | 615 (11.4) | 71 (10.6) | 68 (10.1) | 266 (36.3) | 1020 (13.7) |
| Total Number of Images | | 5374 | 672 | 671 | 732 | 7449 |

Lamps, Rare Lamps together as Lamps which we refer to as *Reduced*. For damage type localization, we merged semantically similar and rare classes together to create a more balanced class distribution which we also refer to as *Reduced*. We additionally explored a further merging of damage types, where we merged damage types label as surface damage, deformity, and body damage which we refer to as *Categorical*. Finally, we also experimented with a coarse no damage versus damage classification, which we refer to as *Binary*. See Table III for complete list of label merging strategies.

### B. Validation Performance for Different Merge Strategies

We trained both joint and independent localization model with all combinations of merge strategies with the same approach mentioned in Section II-C. We used the best performing model on the validation set for further evaluation.

Coarser taxonomy outperforms fine-grained taxonomy across the board (Table IV). On average across all combinations, vehicle parts localization performance increased from 0.709 to 0.798 mIoU for *Original* and *Reduced* taxonomy respectively whereas the damage types localization performance increase from 0.262 to 0.342 to 0.465 to 0.681 for *Original*, *Reduced*, *Categorical*, and *Binary* taxonomy respectively. A coarser vehicle part taxonomy also improved damage type localization performance in the joint model. On average across different damage type taxonomies, the mIoU increased by 0.010. Damage types localization with the joint model outperformed the independent models across all taxonomies whereas vehicle parts localization with independent models outperformed the joint model across all taxonomies.

We chose to use "Reduced" Vehicle Part taxonomy, and "Categorical" Damage Type taxonomy as our final taxonomy to strike a balance between model performance and information loss, i.e. a coarser label is less informative compared to a fine-grained label. Within a continuously running service, label granularity could be changed dynamically based on model performance and data composition. For example, when user has submitted sufficient data for very rare classes such as corrosion and paint chips, the model could be retrained with a more fine-grained taxonomy.

### C. Models and Loss Function

In our initial investigation, we used a subset of the training set (35%, 1900 examples) to assess different segmentation architectures and backbone encoders. We trained a joint localization model with original vehicle part taxonomy and reduced damage types taxonomy using the same approach mentioned in Section II-C, except we fixed the learning rate at 0.0001.

DeepLabV3+ consistently outperformed UNet++ and PSPNet with the same encoder (Table V). On average, vehicle part localization performance increased from 0.415 to 0.539 to 0.595 mIoU for PSPNet, UNet++, and DeepLabV3 respectively whereas damage types localization performance increased from 0.246 to 0.283 to 0.305 respectively. Within the same segmentation architecture, vehicle part localization performance increased from 0.485 to 0.505 to 0.515 to 0.527 to 0.533 to 0.550 for ResNet18, ResNet50, ResNetXt50_32 × 4d, EfficientNet-b3, MiT-b2 and EfficientNet-b5 respectively

TABLE III

DIFFERENT LABEL MERGING STRATEGIES USED IN EARLY EXPERIMENTS. THE FINAL SELECTED TAXONOMY USED REDUCED FOR VEHICLE PART LOCALIZATION AND CATEGORICAL FOR DAMAGE LOCALIZATION

| Merge Strategy | Original | Reduced | Categorical | Binary |
|---|---|---|---|---|
| Vehicle Part Localization | Background | | Background | |
| | Front Bumper | | Bumper | |
| | Rear Bumper | | | |
| | Grille | | Grille | |
| | Front Lamps | | Lamps | |
| | Fog Lamps | | | |
| | Rear Lamps | | | |
| | Front Fender | | Fender | |
| | Windshield | | Windshield | |
| | Front Door Shell | | Door Shell | |
| | Rear Door Shell | | | |
| | Hood | | Hood | |
| | Wheel | | Wheel | |
| | Door Glass | | Window | |
| | Quarter Glass | | | |
| | Back Window | | | |
| | Roof | | Roof | |
| | Mirror | | Mirror | |
| | Windshield Pillar | | Panel | |
| | Rocker Panel | | | |
| | Rear Quarter Panel | | | |
| | Lower Rear Cover | | | |
| | Trunk | | Trunk | |
| Damage Localization | No Damage | No Damage | No Damage | No Damage |
| | Paint Chip | Scratch | Surface Damage | Damage |
| | Scratch | | | |
| | Dent | Dent | Deformity | |
| | Crack | Crack | | |
| | Corrosion | Other | Body Damage | |
| | Missing | | | |
| | Broken | Broken | | |

TABLE IV

PERFORMANCE ON DIFFERENT MERGE STRATEGY COMBINATIONS ON THE VALIDATION SET

| Model Types | Vehicle Parts Label | Damage Types Label | Vehicle Parts mIoU | Damage Types mIoU |
|---|---|---|---|---|
| Joint | Original | Original | 0.712 | 0.262 |
| Joint | Original | Reduced | 0.711 | 0.342 |
| Joint | Original | Categorical | 0.708 | 0.465 |
| Joint | Original | Binary | 0.699 | 0.670 |
| Joint | Reduced | Original | 0.787 | 0.274 |
| Joint | Reduced | Reduced | 0.802 | 0.343 |
| Joint | Reduced | Categorical | 0.793 | 0.472 |
| Joint | Reduced | Binary | 0.798 | 0.689 |
| Independent | Original | N/A | 0.717 | N/A |
| Independent | Reduced | N/A | 0.808 | N/A |
| Independent | N/A | Original | N/A | 0.251 |
| Independent | N/A | Reduced | N/A | 0.340 |
| Independent | N/A | Categorical | N/A | 0.458 |
| Independent | N/A | Binary | N/A | 0.685 |

whereas damage type localization performance increased from 0.2617 to 0.262 to 0.274 to 0.28 to 0.292 to 0.299 for ResNet18, ResNet50, ResNetXt50_32 × 4d, MiT-b2, EfficientNet-b3 and EfficientNet-b5 respectively.

TABLE V

PERFORMANCE OF DIFFERENT BACKBONE ENCODERS AND SEGMENTATION ARCHITECTURES ON THE VALIDATION SET

| Segmentation Architecture | Encoder | Vehicle Parts mIoU | Damage Types mIoU |
|---|---|---|---|
| DeepLabV3+ | EfficientNet-b5 | 0.638 | 0.331 |
| DeepLabV3+ | EfficientNet-b3 | 0.607 | 0.323 |
| DeepLabV3+ | MiT-b2 | 0.586 | 0.299 |
| DeepLabV3+ | ResNetXt50_32x4d | 0.585 | 0.301 |
| DeepLabV3+ | ResNet50 | 0.583 | 0.277 |
| DeepLabV3+ | ResNet18 | 0.564 | 0.293 |
| UNet++ | EfficientNet-b5 | 0.598 | 0.310 |
| UNet++ | EfficientNet-b3 | 0.609 | 0.317 |
| UNet++ | MiT-b2 | 0.583 | 0.293 |
| UNet++ | ResNetXt50_32x4d | 0.506 | 0.272 |
| UNet++ | ResNet50 | 0.507 | 0.266 |
| UNet++ | ResNet18 | 0.473 | 0.248 |
| PSPNet | EfficientNet-b5 | 0.413 | 0.256 |
| PSPNet | EfficientNet-b3 | 0.364 | 0.237 |
| PSPNet | ResNetXt50_32x4d | 0.454 | 0.248 |
| PSPNet | MiT-b2 | 0.431 | 0.248 |
| PSPNet | ResNet50 | 0.425 | 0.243 |
| PSPNet | ResNet18 | 0.419 | 0.244 |

TABLE VI

PERFORMANCE OF DIFFERENT LOSS FUNCTIONS ON THE VALIDATION SET

| Loss Function | Vehicle Parts mIoU | Damage Types mIoU |
|---|---|---|
| Dice | 0.607 | 0.323 |
| Cross Entropy | 0.560 | 0.313 |
| Focal | 0.555 | 0.303 |

We also experimented with cross entropy loss, dice loss, and focal loss with EfficientNet-b3 as the backbone encoder and DeepLabV3+ as the segmentation architecture and using the above mentioned training procedure. Dice loss consistently outperformed both cross entropy and focal loss (Table VII).

### D. Class Imbalance

We explored the use of undersampling and oversampling to address the class imbalance in the damage type classes. We first computed the inverse of damage type class frequency or the ratio of the number of samples in class to the total number of samples. Then to perform undersampling, we sampled a number of instances equal to the minimum class size multiplied by the number of classes without replacement. To perform oversampling, we sampled a total count that's equal to the maximum class size multiplied by the number of classes with replacement.

For the undersampling strategy, we found vehicle parts localization performance decreased from 0.712 to 0.642 and

TABLE VII
PERFORMANCE OF DIFFERENT DATASET SAMPLING STRATEGY ON THE VALIDATION SET

| Sampling Strategy | Vehicle Parts mIoU | Damage Types mIoU |
|---|---|---|
| Under-sampling | 0.642 | 0.194 |
| Over-sampling | 0.713 | 0.261 |
| None | 0.712 | 0.262 |

damage types localization performance decreased from 0.262 to 0.194. For the oversampling strategy, we found vehicle parts localization performance of 0.713 and damage types localization performance of 0.261, which was similar to the original model performance.

## REFERENCES


[1] A. Shirode, T. Rathod, P. Wanjari, and A. Halbe, "Car damage detection and assessment using CNN," in *Proc. IEEE Delhi Sect. Conf. (DELCON)*, Feb. 2022, pp. 1–5, doi: 10.1109/DELCON54057.2022.9752971.

[2] A. C. Chua et al., "Damage identification of selected car parts using image classification and deep learning," in *Proc. IEEE 13th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ., Manage. (HNICEM)*, Nov. 2021, pp. 1–5, doi: 10.1109/HNICEM54116.2021.9731806.

[3] U. Waqas, N. Akram, S. Kim, D. Lee, and J. Jeon, "Vehicle damage classification and fraudulent image detection including Moiré effect using deep learning," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, Aug. 2020, pp. 1–5, doi: 10.1109/CCECE47787.2020.9255806.

[4] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, Feb. 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125, doi: 10.3390/info11020125.

[5] C. G. Pachón-Suescún, P. C. U. Murillo, and R. Jimenez-Moreno, "Scratch detection in cars using a convolutional neural network by means of transfer learning," *Int. J. Appl. Eng. Res.*, vol. 13, no. 16, pp. 12976–12982, 2018.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[7] D. R. Cox, "The regression analysis of binary sequences," *J. Roy. Stat. Soc. B, Methodol.*, vol. 21, no. 1, p. 238, Jan. 1959.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[9] H. Bandi, S. Joshi, S. Bhagat, and A. Deshpande, "Assessing car damage with convolutional neural networks," in *Proc. Int. Conf. Commun. Inf. Comput. Technol. (ICCICT)*, Jun. 2021, pp. 1–5, doi: 10.1109/ICCICT50803.2021.9510069.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[11] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[12] Y. Huang and Y. Chen, "Autonomous driving with deep learning: A survey of state-of-art technologies," 2020, *arXiv:2006.06091*.

[13] K. Pasupa, P. Kittiworapanya, N. Hongngern, and K. Woraratpanya, "Evaluation of deep learning algorithms for semantic segmentation of car parts," *Complex Intell. Syst.*, vol. 8, no. 5, pp. 3613–3625, Oct. 2022, doi: 10.1007/s40747-021-00397-8.

[14] K. Patil, M. Kulkarni, A. Sriraman, and S. Karande, "Deep learning based car damage classification," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 50–54, doi: 10.1109/ICMLA.2017.0-179.

[15] P. M. Kyu and K. Woraratpanya, "Car damage detection and classification," in *Proc. 11th Int. Conf. Adv. Inf. Technol.*, Jul. 2020, vol. 11, no. 46, pp. 1–6, doi: 10.1145/3406601.3406651.

[16] L. Li, K. Ono, and C.-K. Ngan, "A deep learning and transfer learning approach for vehicle damage detection," in *Proc. Int. FLAIRS Conf.*, vol. 34, 2021, pp. 1–6.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Jan. 2022.

[19] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4CV: A photo-realistic simulator for computer vision applications," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 902–919, Sep. 2018.

[20] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "A very deep transfer learning model for vehicle damage detection and localization," in *Proc. 31st Int. Conf. Microelectron. (ICM)*, Dec. 2019, pp. 158–161, doi: 10.1109/ICM48031.2019.9021687.

[21] K. Ohri and M. Kumar, "Review on self-supervised image recognition using deep neural networks," *Knowl.-Based Syst.*, vol. 224, Jul. 2021, Art. no. 107090.

[22] R. Prabhu and K. Srilekha, "Automating vehicle damage estimation with computer vision," in *Proc. GTC Digit. Spring*, 2022, pp. 12–22.

[23] Q. Zhang, X. Chang, and S. B. Bian, "Vehicle-damage-detection segmentation algorithm based on improved mask RCNN," *IEEE Access*, vol. 8, pp. 6997–7004, 2020, doi: 10.1109/ACCESS.2020.2964055.

[24] A. Radford, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[25] P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol, "AI in health and medicine," *Nature Med.*, vol. 28, no. 1, pp. 31–38, 2022.

[26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10684–10695.

[27] C. M. Sruthy, S. Kunjumon, and R. Nandakumar, "Car damage identification and categorization using various transfer learning models," in *Proc. 5th Int. Conf. Trends Electron. Informat. (ICOEI)*, Jun. 2021, pp. 1097–1101, doi: 10.1109/ICOEI51242.2021.9452846.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[30] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Québec City, QC, Canada: Springer, 2017, pp. 240–248.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015, *arXiv:1512.00567*.

[33] H. Tercan and T. Meisen, "Machine learning and deep learning based predictive quality in manufacturing: A systematic review," *J. Intell. Manuf.*, vol. 33, no. 7, pp. 1879–1905, Oct. 2022.

[34] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

[35] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 8934–8954, Sep. 2023.

[36] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[37] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Granada, Spain: Springer, 2018, pp. 3–11.

[38] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

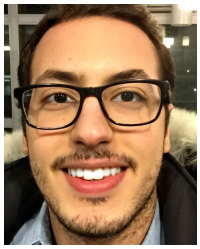
**Yuntao Ma** is currently pursuing the master's degree in electrical engineering with Stanford University, with a focus on control and optimization. His primary research interests include deep learning in computer vision.

**Hiva Ghanbari** received the Ph.D. degree in industrial and systems engineering from Lehigh University, Bethlehem, PA, USA, in 2018, with a focus on convex optimization in machine learning problems. She is currently a Senior AI Specialist with the AI Advancement Center, Ford Motor Company.

**Tianyuan Huang** is currently pursuing the Ph.D. degree with Stanford University. His research interests include machine learning applications in urban sustainability–particularly applying computer vision in the spatial-temporal context, such as measuring and mapping urban change using historical street view and satellite images.

**Jeremy Irvin** is currently pursuing the Ph.D. degree with Stanford University. His research interests include developing computer vision techniques for social good applications, including medicine, reforestation, renewable energy, and transportation.

**Oliver Brady** received the master's degree in computer science from Stanford University, with a focus on machine learning. His primary research interests include computer vision applications for climate change solutions.



**Sofian Zalouk** is currently pursuing the master's degree in computer science with Stanford University. His primary research interests include developing machine learning techniques for improving fairness and for social good applications, including healthcare and transportation.

**Hao Sheng** received the Ph.D. degree in computer engineering from Stanford University. He is currently a Machine Learning Engineer with the Apple Special Projects Group, with a focus on developing data-centric deep learning approaches.

**Andrew Ng** is currently the Founder of DeepLearning.AI, the Founder and the CEO of Landing.AI, a General Partner with AI Fund, the Chairperson and the Co-Founder of Coursera, and an Adjunct Professor with Stanford University.

**Ram Rajagopal** (Member, IEEE) is currently an Associate Professor of civil and environmental engineering with Stanford University. He is the Founder and the Director of the Stanford Sustainable Systems Laboratory. He has also worked extensively on sensing infrastructure systems and transportation networks. His primary research interests include advancing the design, optimization, and data-driven modeling of electric power systems.

**Mayur Narsude** received the Ph.D. degree in physics from École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2014, with a focus on functional magnetic resonance imaging. He is currently a Manager of computer vision with the AI Advancement Center, Ford Motor Company.