CrashCar101: Procedural Generation for Damage Assessment - Supplementary Materials

Jens Parslov*,[1] Erik Riise*,[1] Dim P. Papadopoulos[1,2]
[1] Technical University of Denmark [2] Pioneer Center for AI
jens@parslov.com, erikriise@live.no, dimp@dtu.dk
https://crashcar.compute.dtu.dk

The purpose of the supplementary material is to provide additional information about the creation of the CrashCar101 dataset. We further show more qualitative results for both part and damage segmentation tasks, more dataset statistics as well as more example images and annotations.

1. Dataset statistics

To gain an improved overview of CrashCar101, we provide some statistics about the CrashCar101 dataset. For comparison, the same statistics are provided for the CarDD dataset. In Fig. 1, we show the proportion of damaged pixels per image and the number of damage types per image. We see that CrashCar101 generally has fewer damaged pixels than CarDD. The number of damages per image shows that the majority of images in CarDD (about 65%) have only one damage. Meanwhile, the images in CrashCar101 have on average more damage types.

Fig. 2 shows the spatial distribution of the damage locations for CrashCar101 and CarDD. We show the distribution of all damages combined as well as the distribution of each damage type. We observe that the spatial distributions of each damage type have some natural bias. For example, glass shatter appears more at the top part of the images because it can only appear on windows, while cracks are not as smooth as the other damage types. We also observe that CrashCar101 has more centered damage compared to CarDD.

2. Part annotation

In Fig. 3, we display 36 examples of 3D car models from ShapeNetCore which we annotated. The 99 cars featured in CrashCar101 are annotated with 27 fine-grained semantic parts. As described in the main paper, the cars were labeled using a human-in-the-loop interactive approach.

*Denotes equal contribution

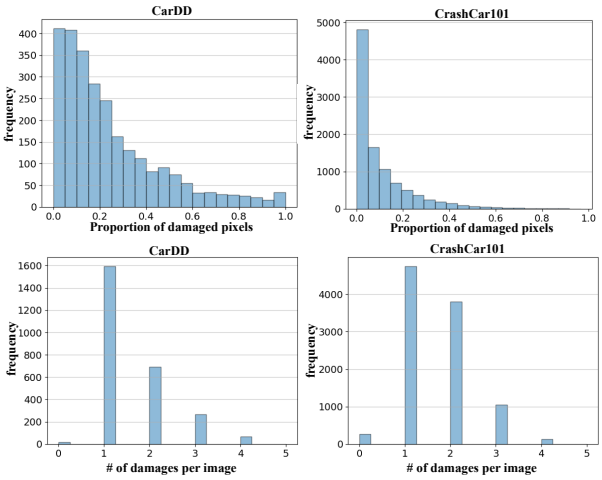


Figure 1. **Dataset Statistics.** (Top row) The proportion of damaged pixels per image. (Bottom row) The number of damage types per image. Both rows show statistics for CarDD and CrashCar101.

3. Part segmentation results

Fig. 4 shows an overview of predictions from a model trained on Pascal-Part, and a DeepLabv3 model trained on our CrashCar101 along with Pascal-Part. Both models use a ResNet50 backbone and were trained without data augmentations. The models trained on CrashCar101 yield better results when testing on either the test data of UDA-PART or Pascal-Part. We observe that the addition of CrashCar101 makes a more precise prediction for mirror part classes.

4. Dent generation details

In Fig. 5 we show the full Geometry Node setup in Blender. The dent map (Fig. 7) outputs a vector field, which is passed to Perturb Mesh which is used to dent the input geometry. The adasub (Fig. 6) node adaptive subdivides the mesh, whilst keeping the original shape. Annotation Material applies a duplicate car paint material to the patch of

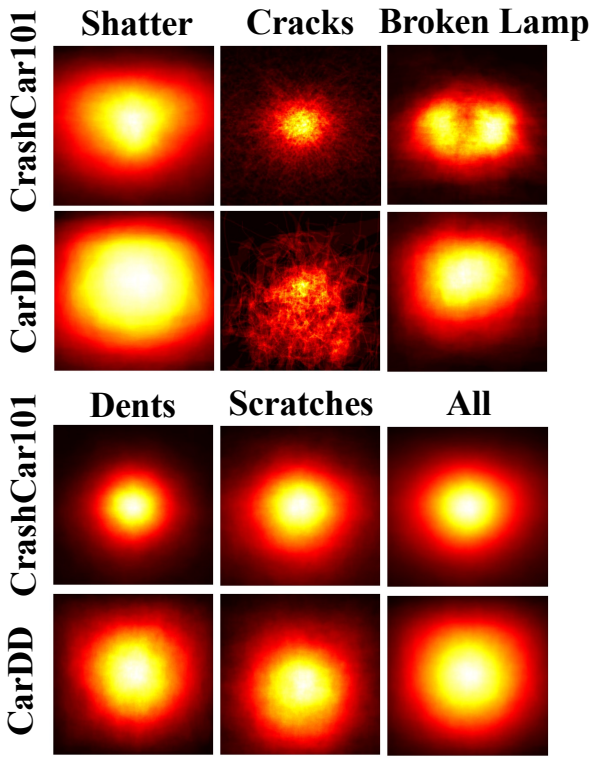


Figure 2. **Damage Locations.** Shows heat maps for image locations of each damage in CrashCar101 and CarDD. Notice that CrashCar101 is more dense in the the center of the image.

the car that has been dented. This annotation material has a unique material ID, which can be subsequently used for 2D annotations. If subdivision is needed, it ensures that only the parts that are inflicted with dents will be subdivided. The dent map is implemented as described in the paper.

5. Blender Shader Nodes

In Fig. 9-13 we show complete implementations of blender shader nodes for scratches, cracks, shatters, and broken lights respectively. The formulas used to produce the damages are explained in the paper.

6. Examples from CrashCar101

The last five pages of the supplementary material are dedicated showing a random selection of various examples from the CrashCar101 dataset. The images show that our synthetic dataset is diverse in viewpoint, car color, background/lighting, and damage appearance, shape, and size. Due to the procedural generation method used in the dataset creation, certain examples can pose a challenge, particularly when defects are situated in shaded areas of the car. Conversely, real data is often obtained under human supervision, ensuring easier visibility of damages, however may also contain bias.

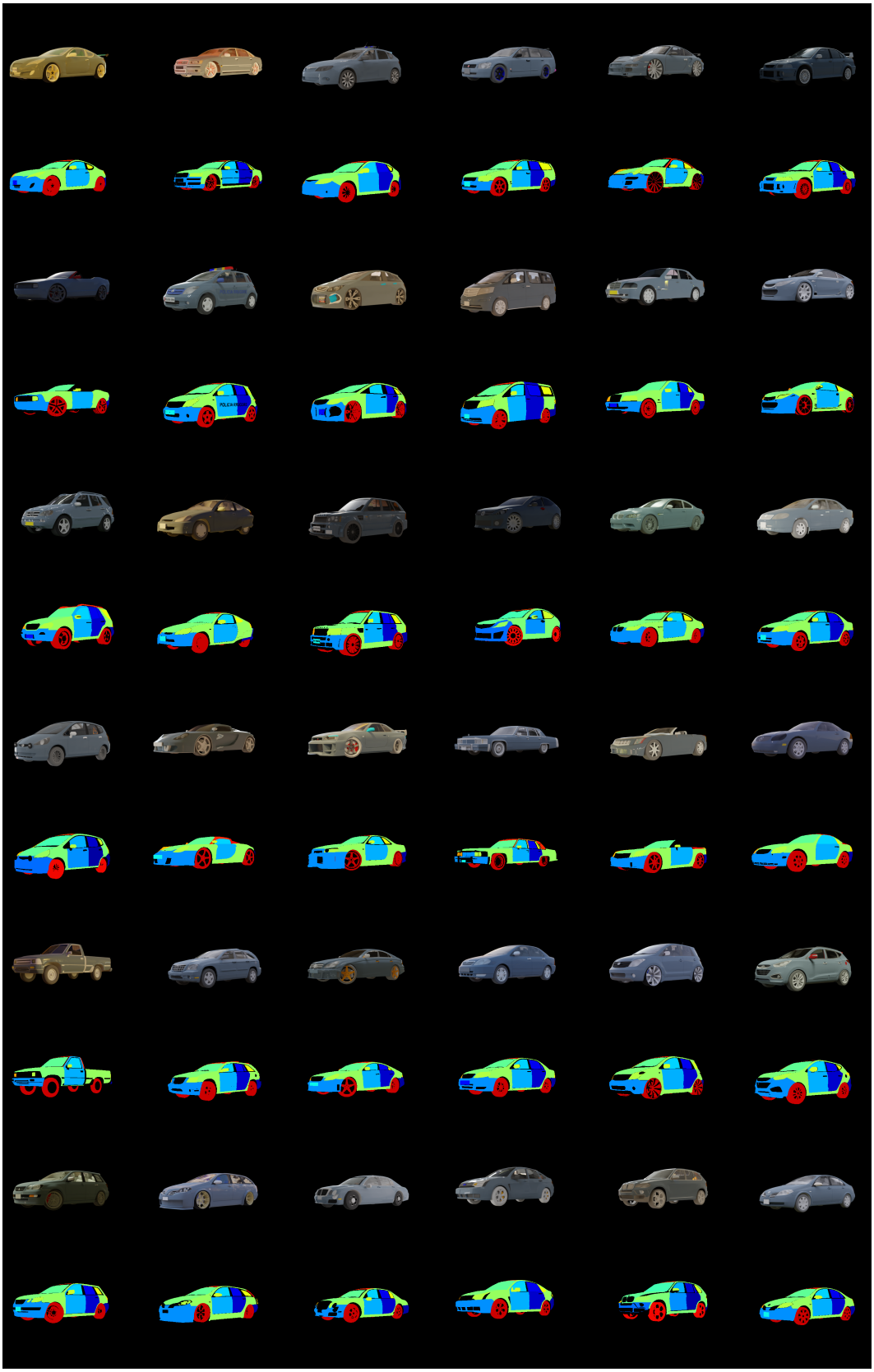


Figure 3. **Examples of 3D models.** We show examples of the models from ShapeNetCore along with the part annotations.

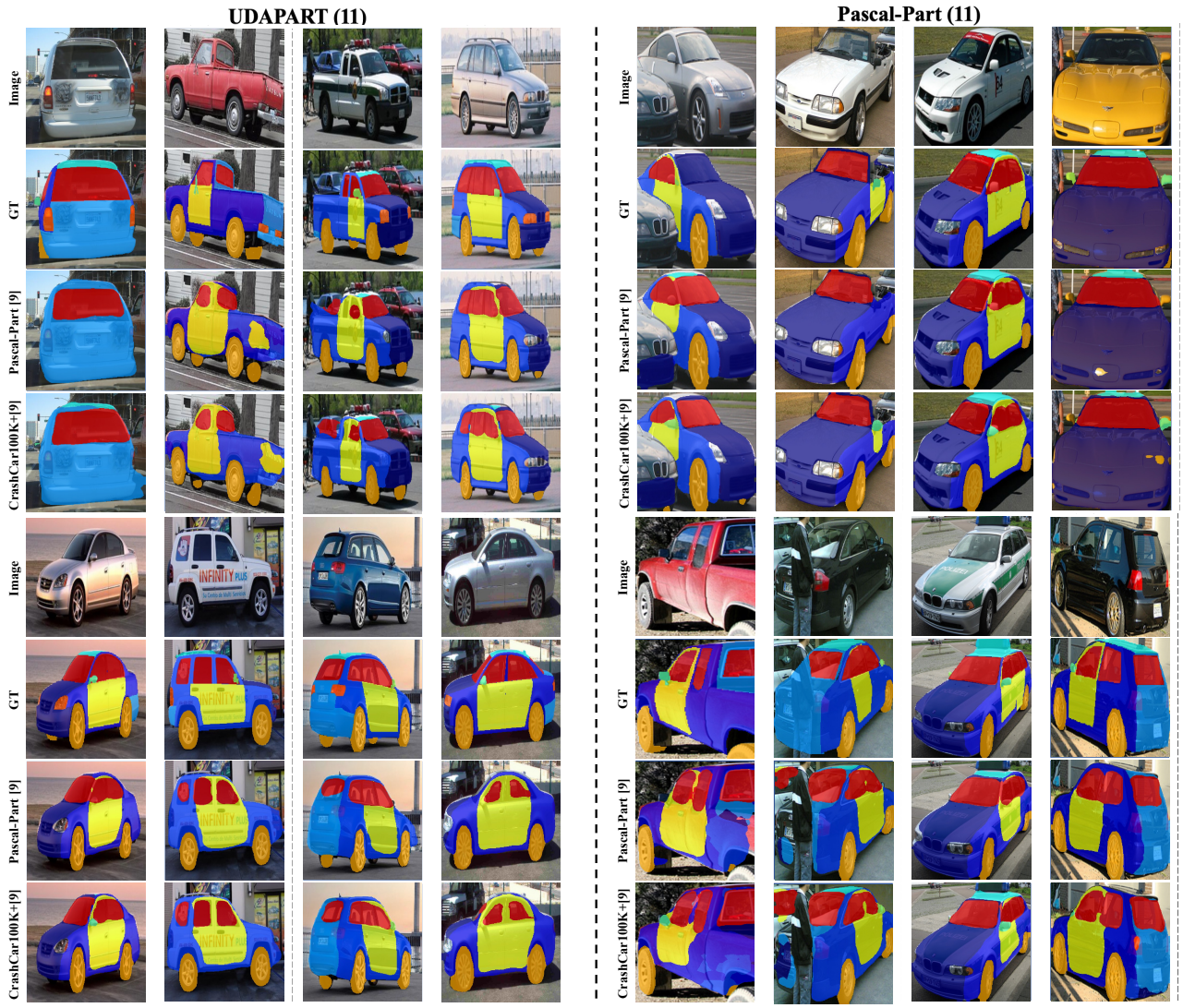


Figure 4. **Qualitative part segmentation results.** We show results from training our part segmentation model on Pascal-Part and on CrashCar101+Pascal-Part. We observe that by including our synthetic data to the real training set, we obtain a model that yields better results.

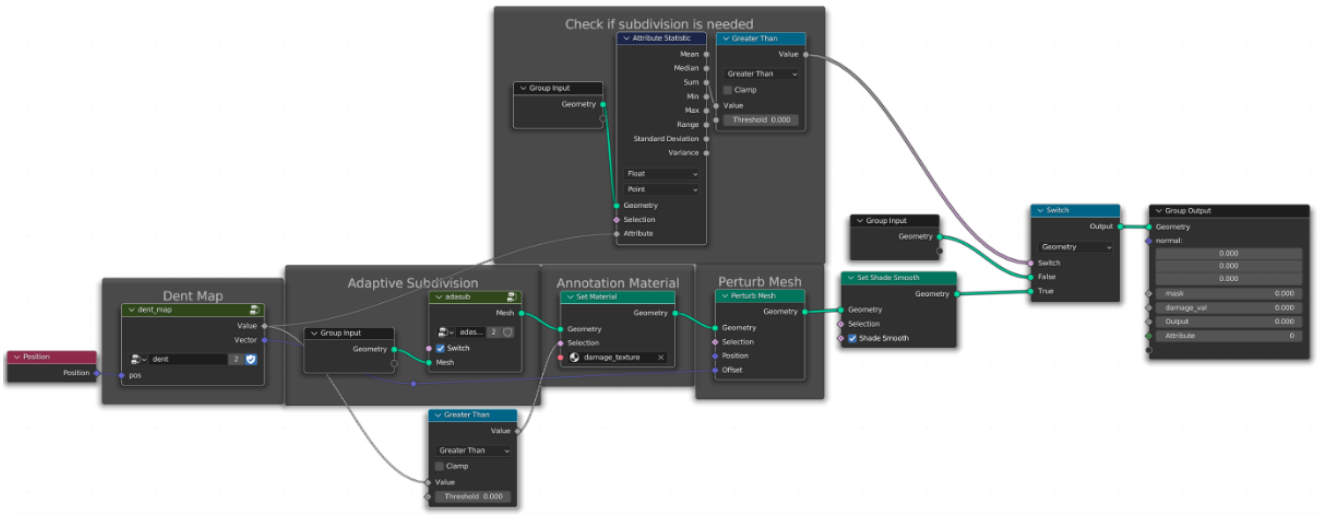


Figure 5. **Geometry Nodes for Dent Generation.** We show the complete implementation of the dent generator.

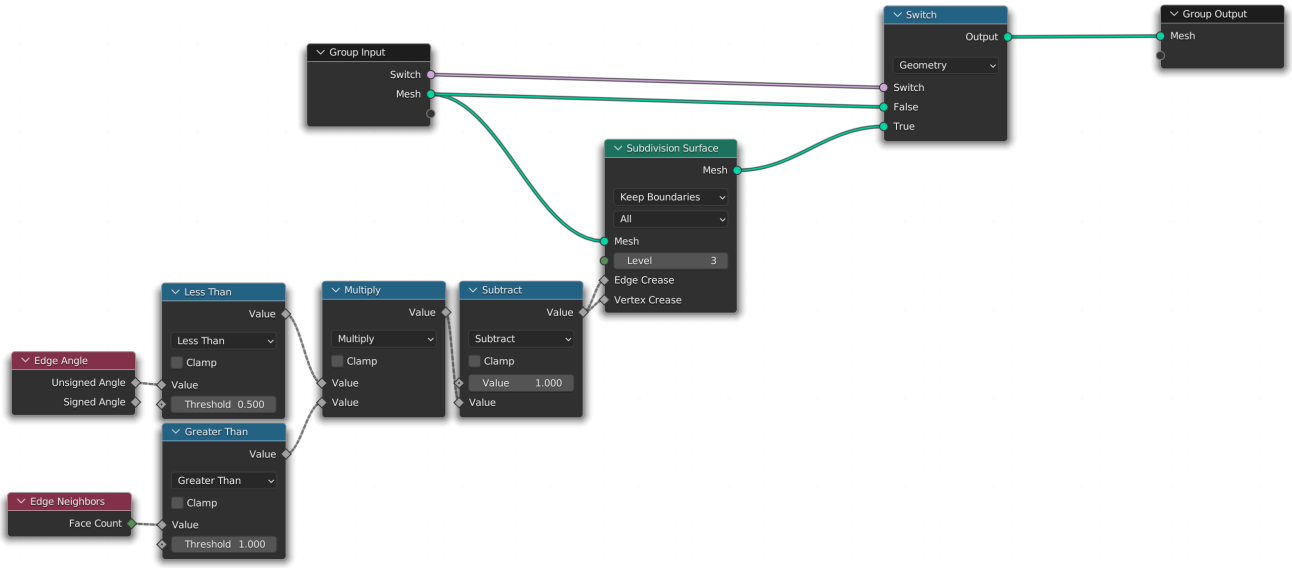


Figure 6. **Adaptive Subdivision.** We show the complete implementation adaptive subdivision.

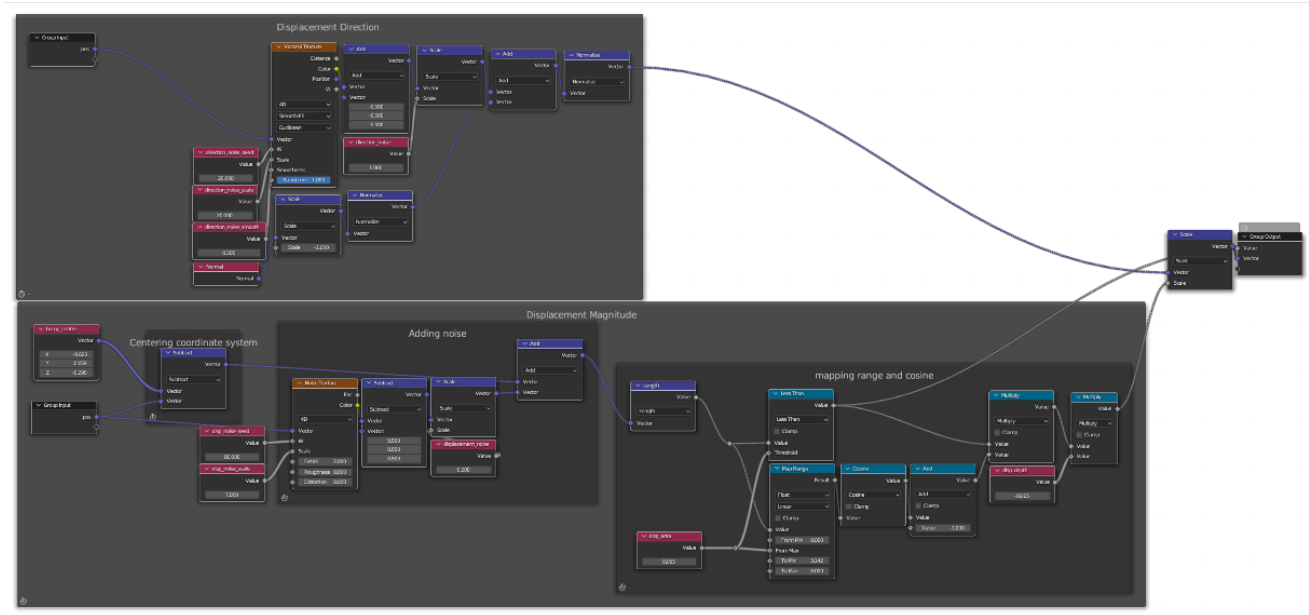


Figure 7. **Dent Map.** This nodetree outputs a vector field used to perturb the input mesh.

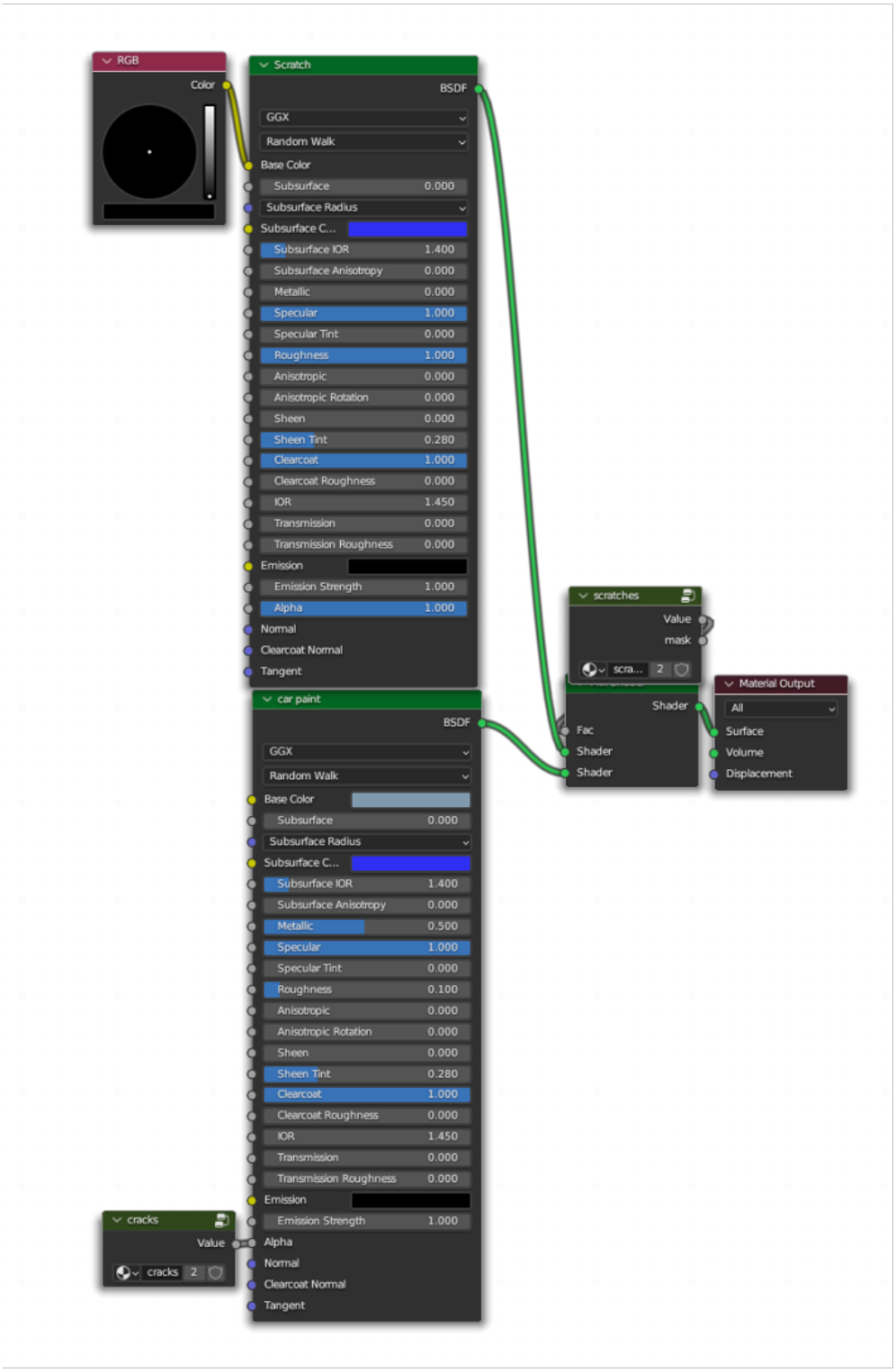


Figure 8. **Damage Texture.** Shows our implementation of scratches and cracks. Cracks drive the alpha value of car paint shader, whilst scratches drives the mix shader node.

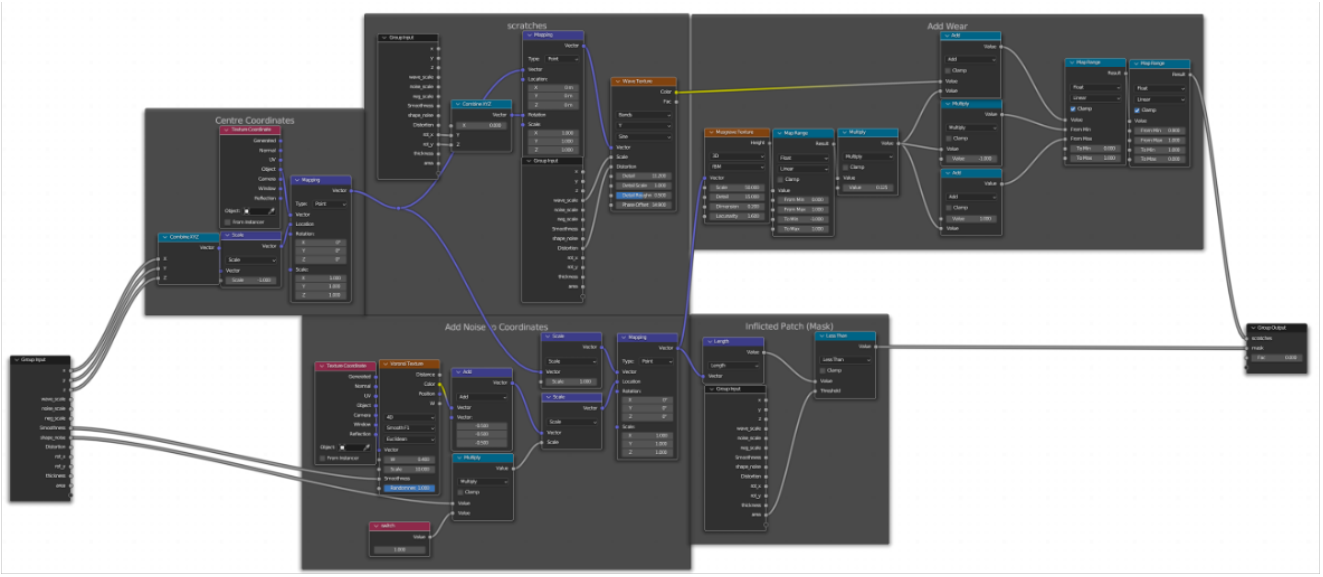


Figure 9. **Scratches.** The output of scratches is a binary map. Patch is used to generate the final segmentation mask annotation.

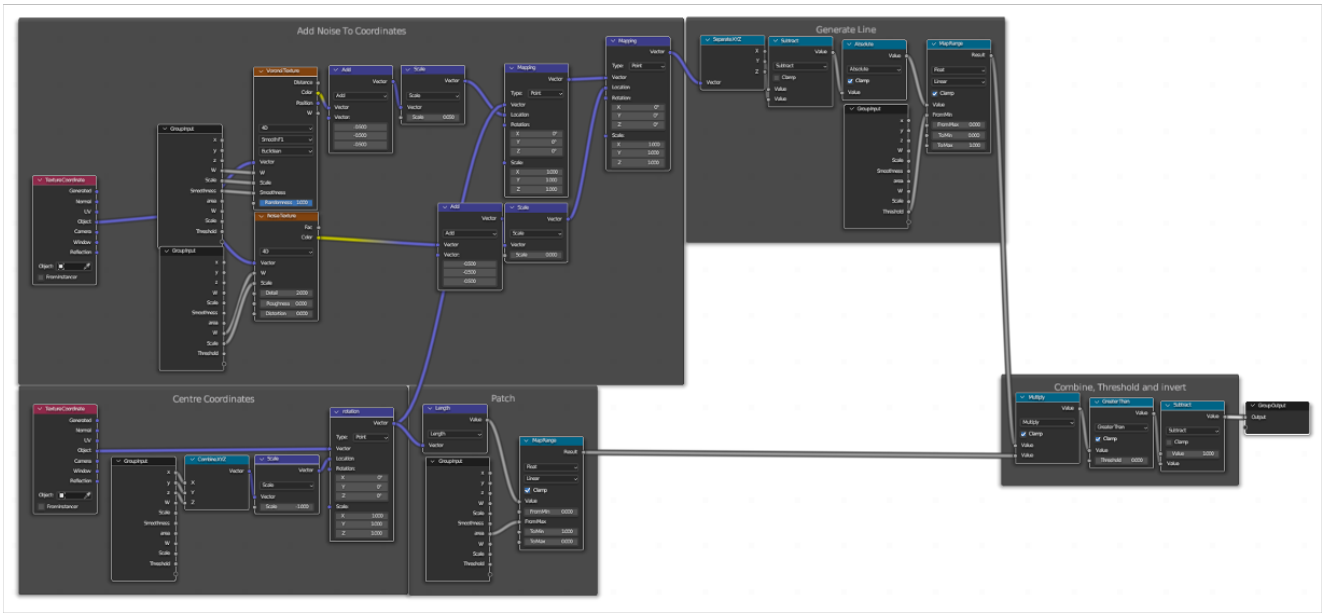


Figure 10. **Cracks.** Shows our implementation of cracks as described in the paper.

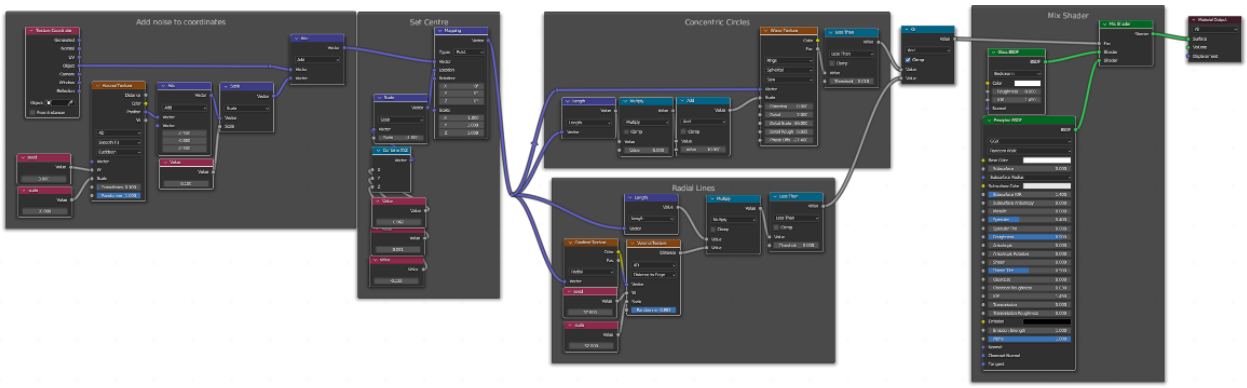


Figure 11. **Shatter Window Shader.** This nodetree implements the shatter glass defect as described in the paper.

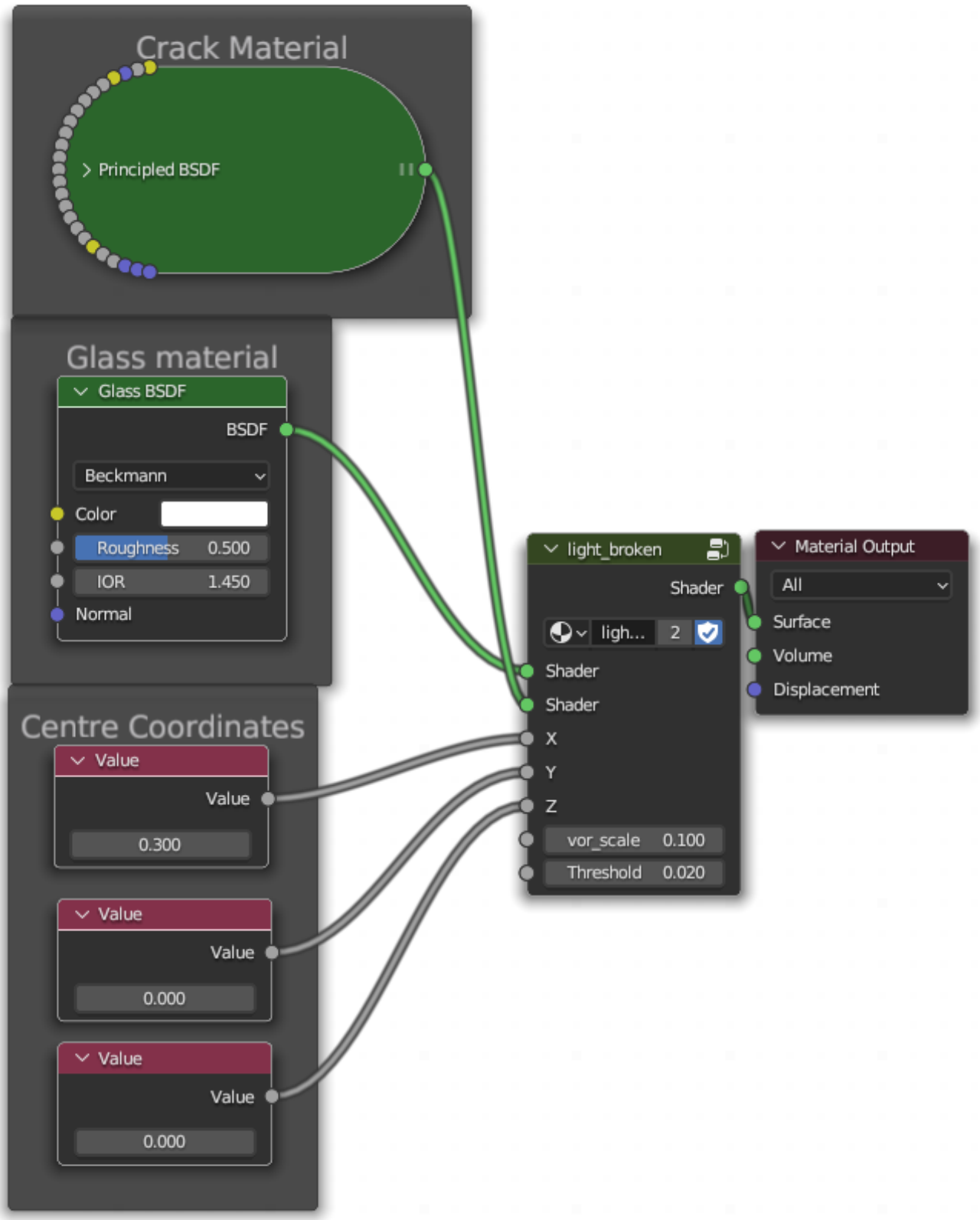


Figure 12. **Light broken parent shader.** This nodetree implements the broken light defect as described in the paper.

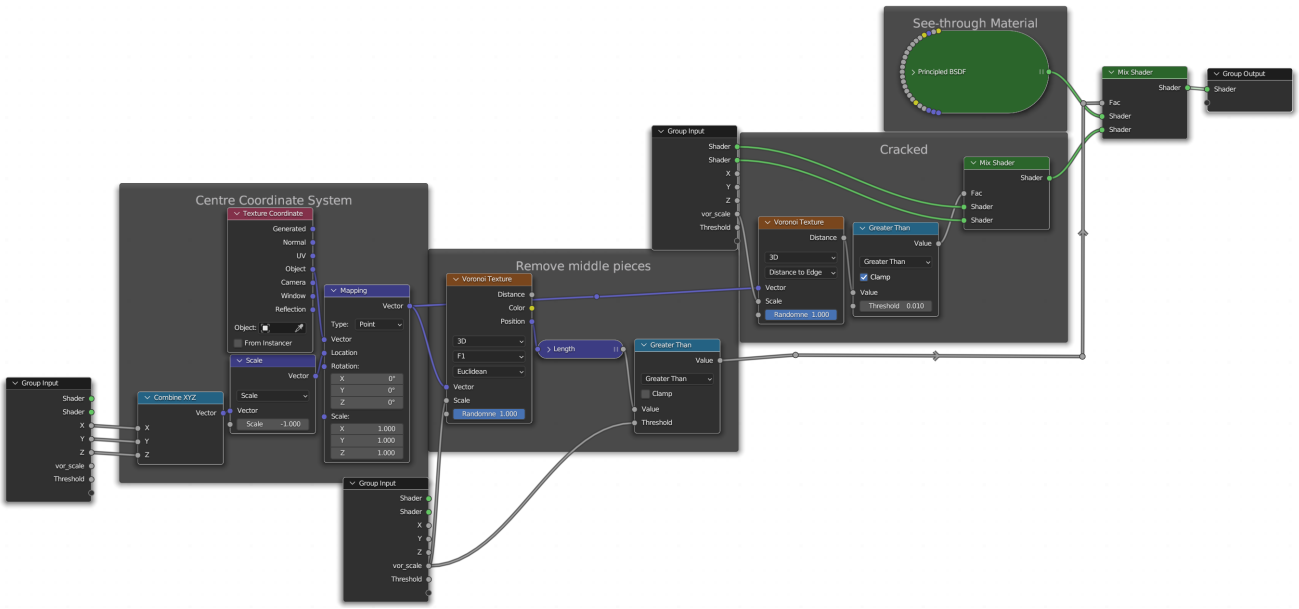


Figure 13. **Light broken parent shader.** This nodetree implements the broken light defect as described in the paper.

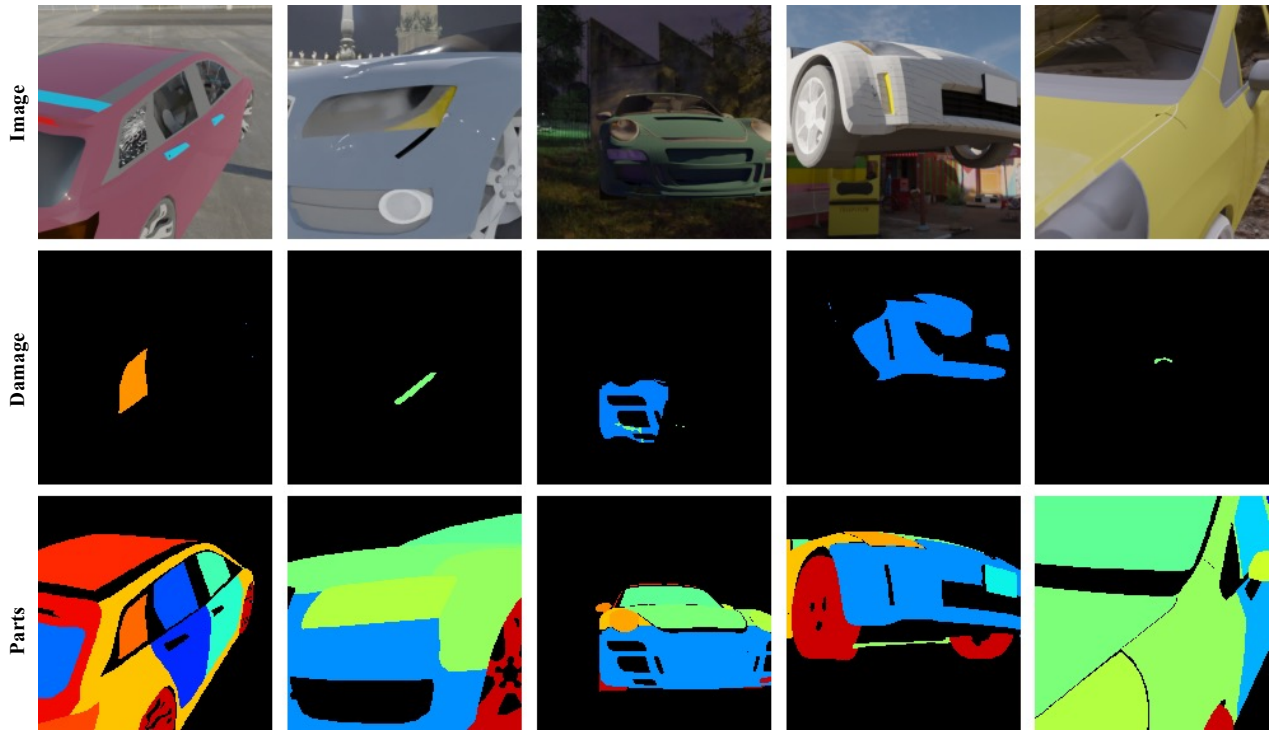


Image
Damage
Parts

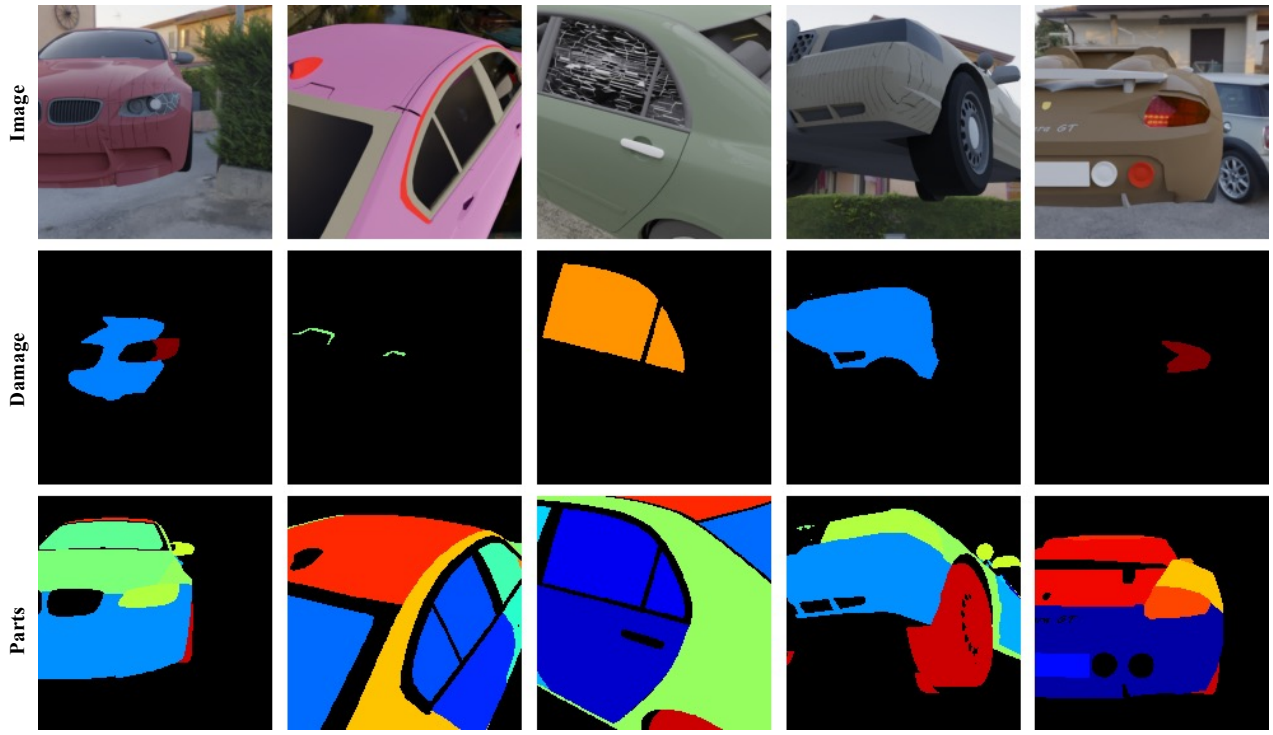
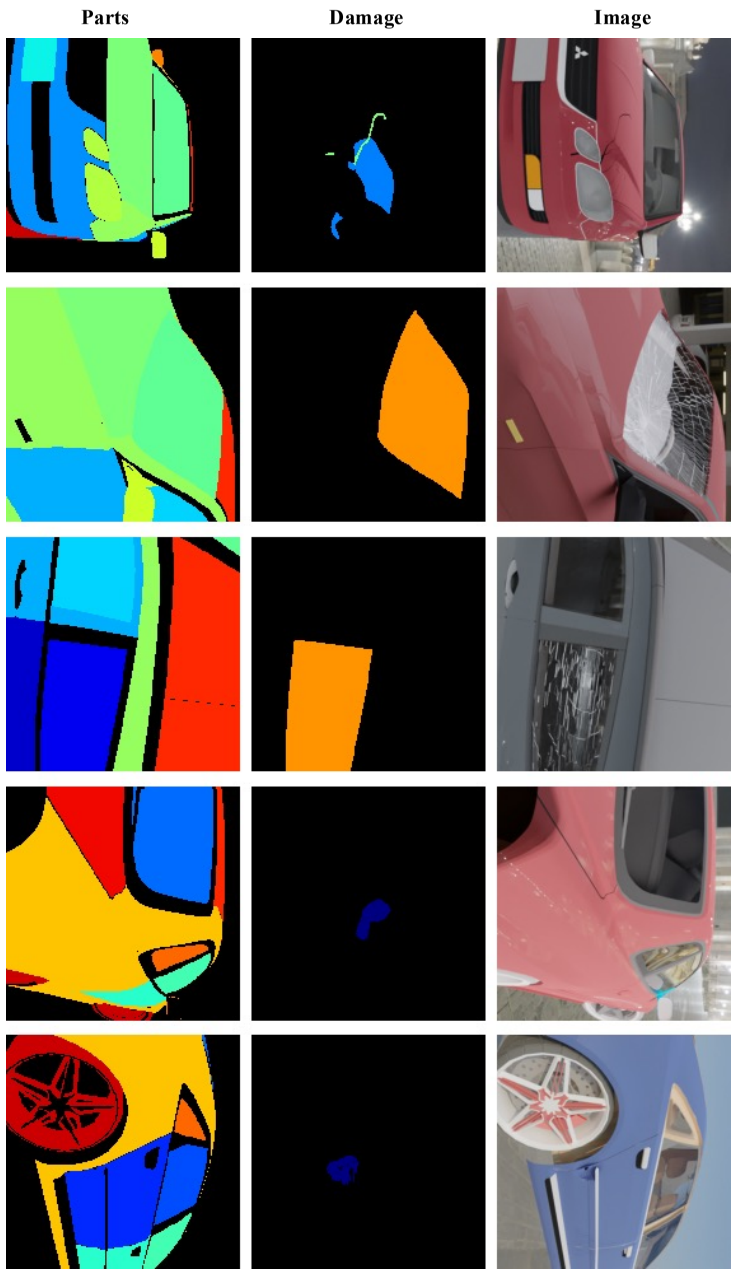
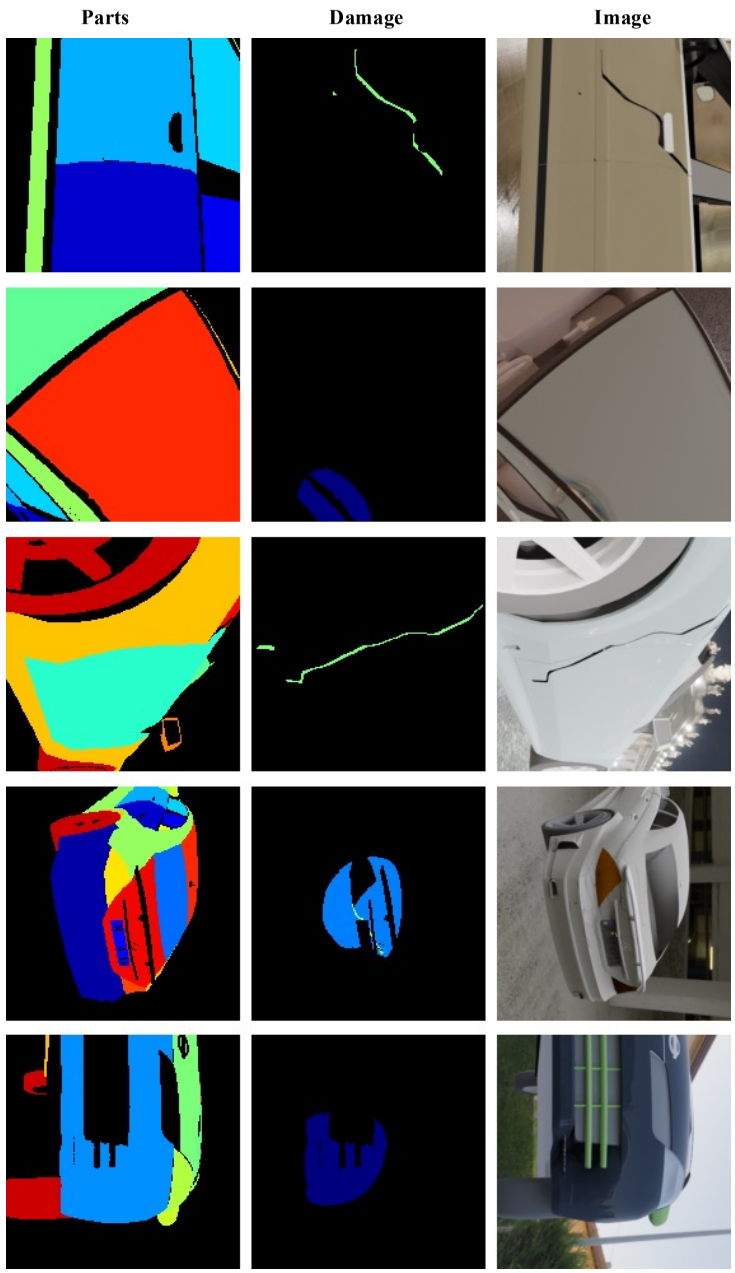
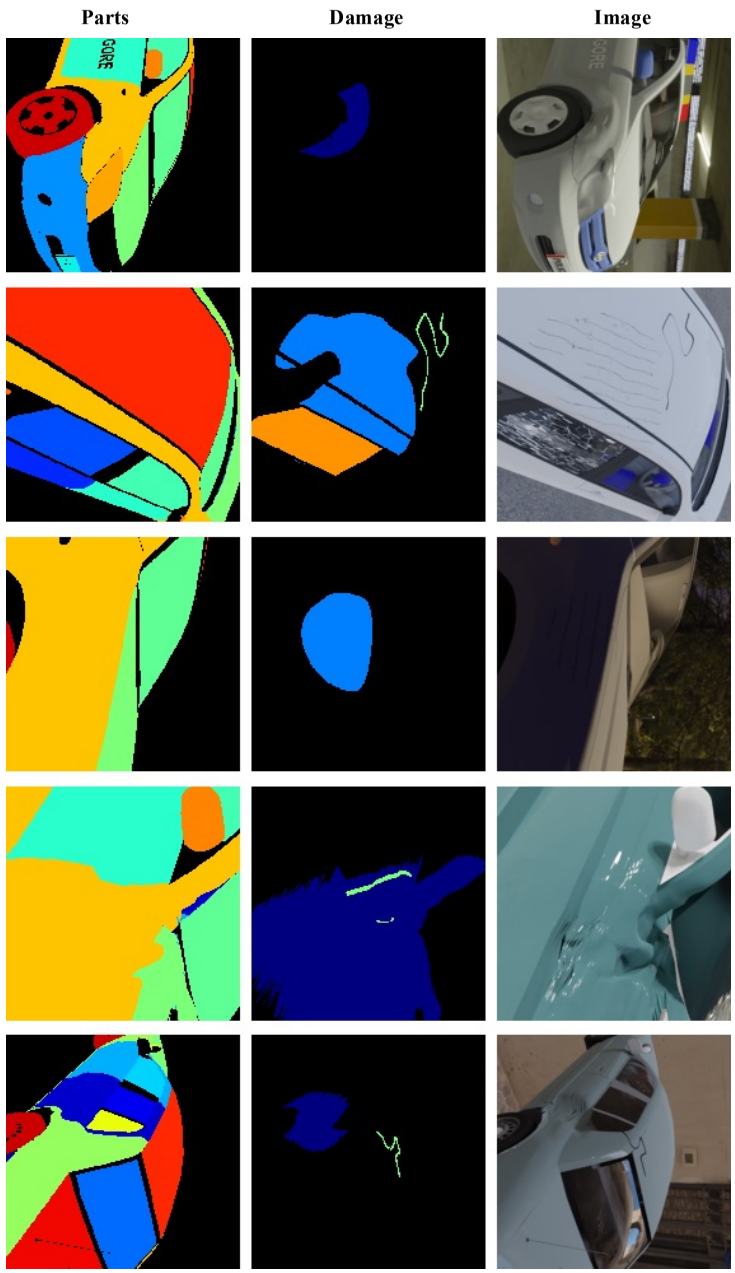
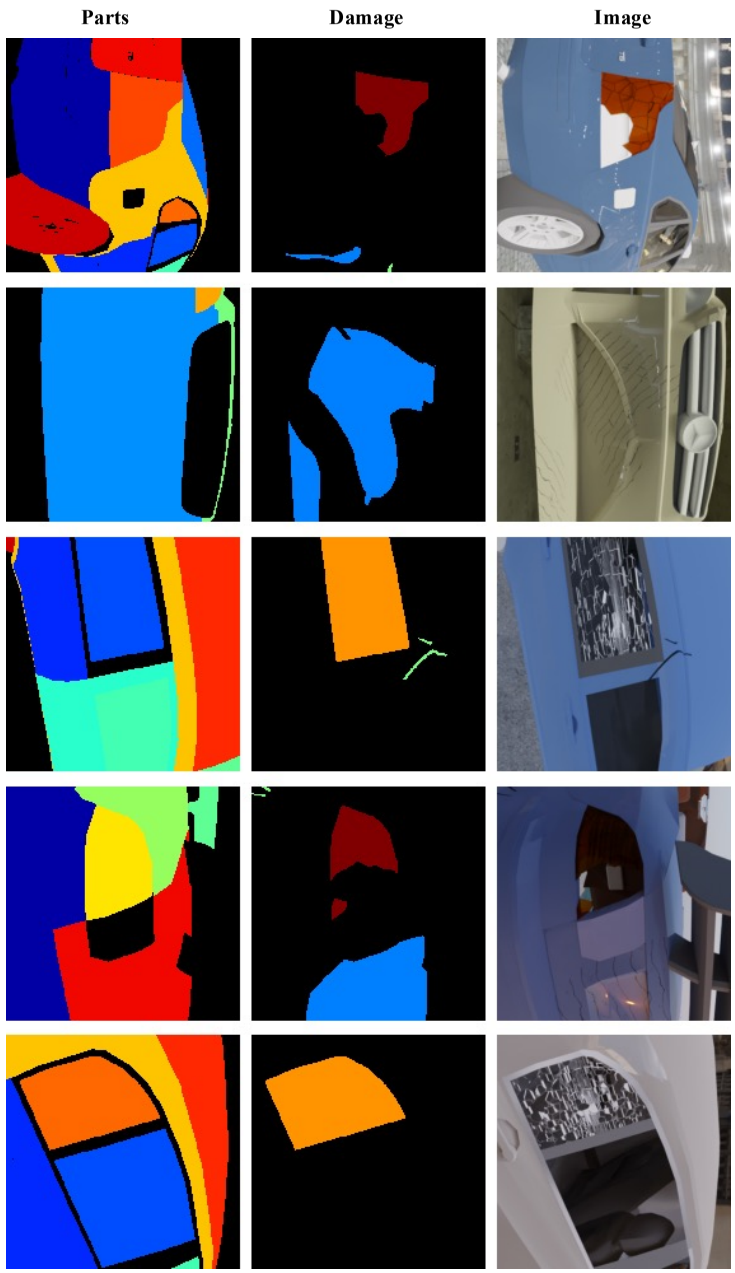


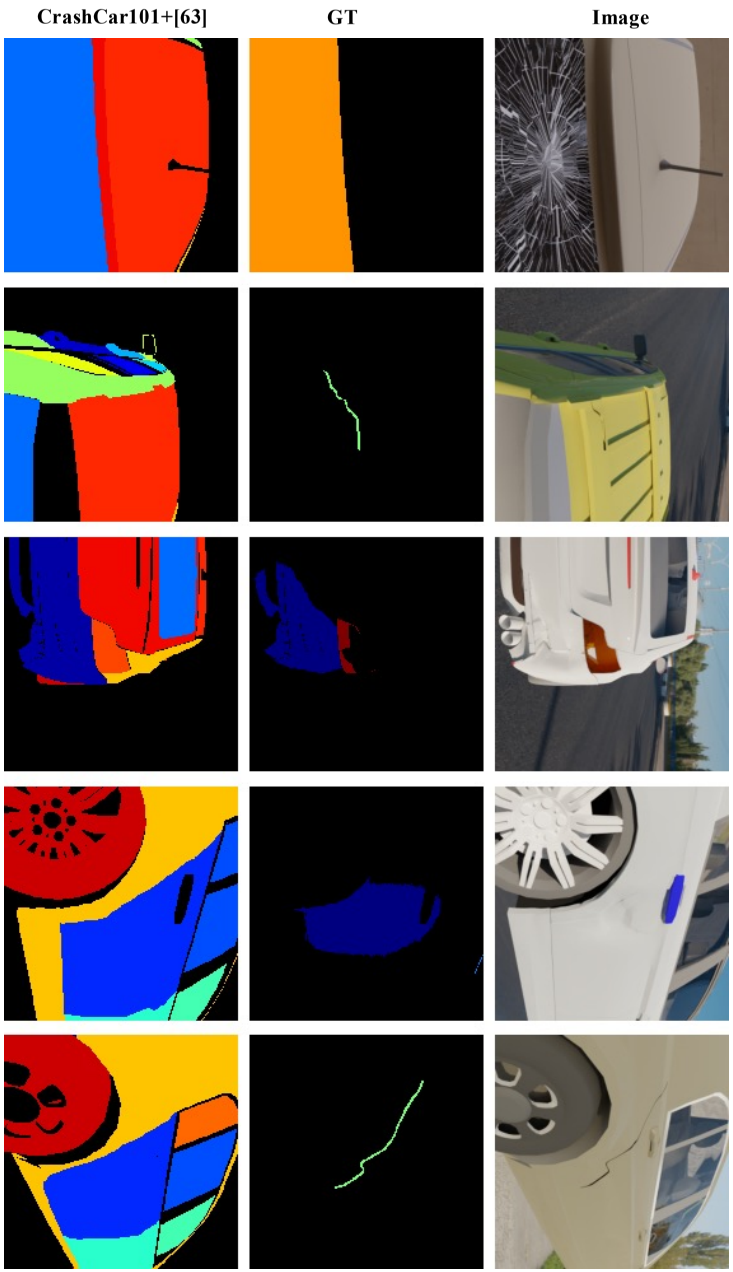
Image
Damage
Parts

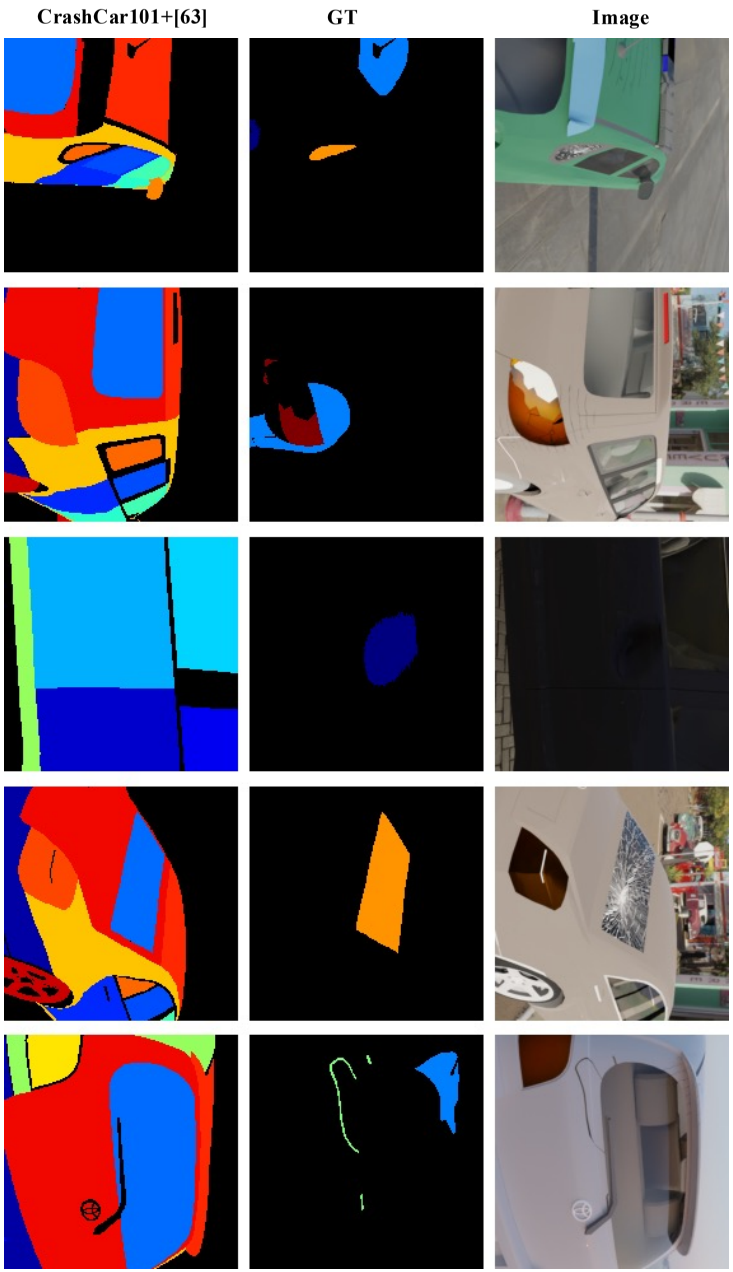

Parts
Damage
Image

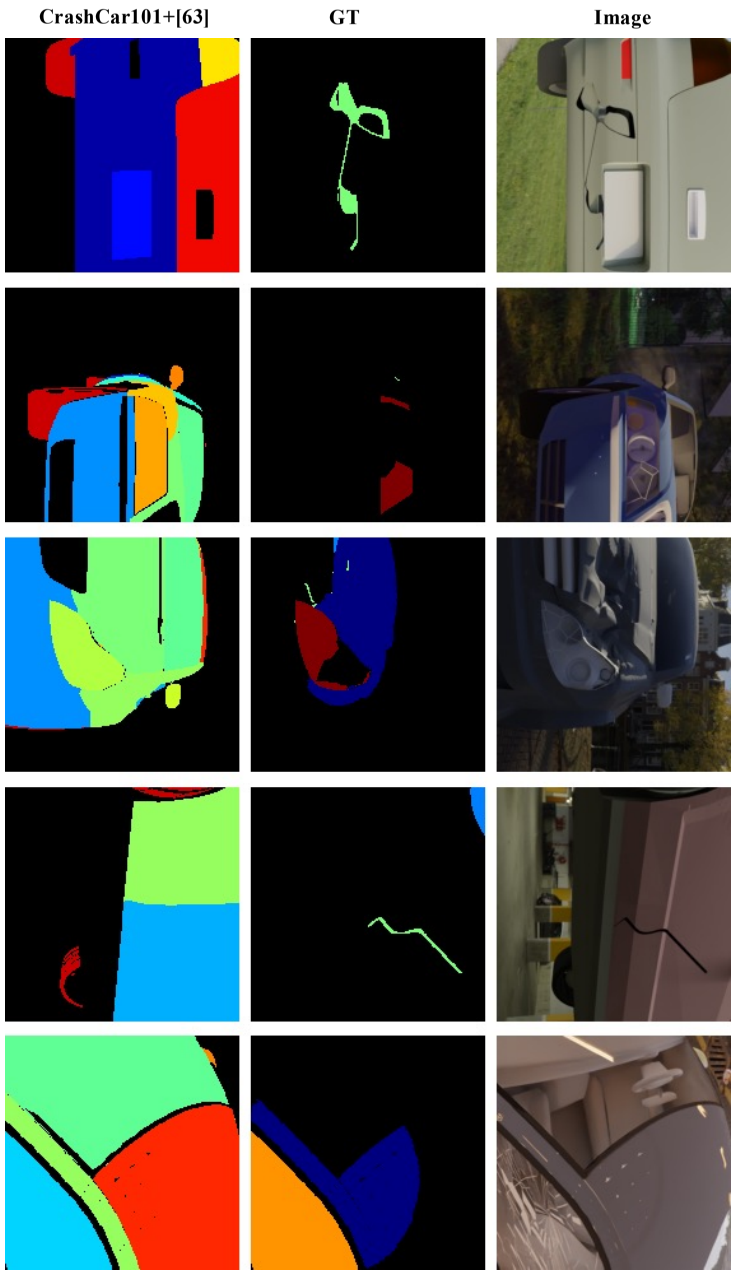

Parts
Damage
Image

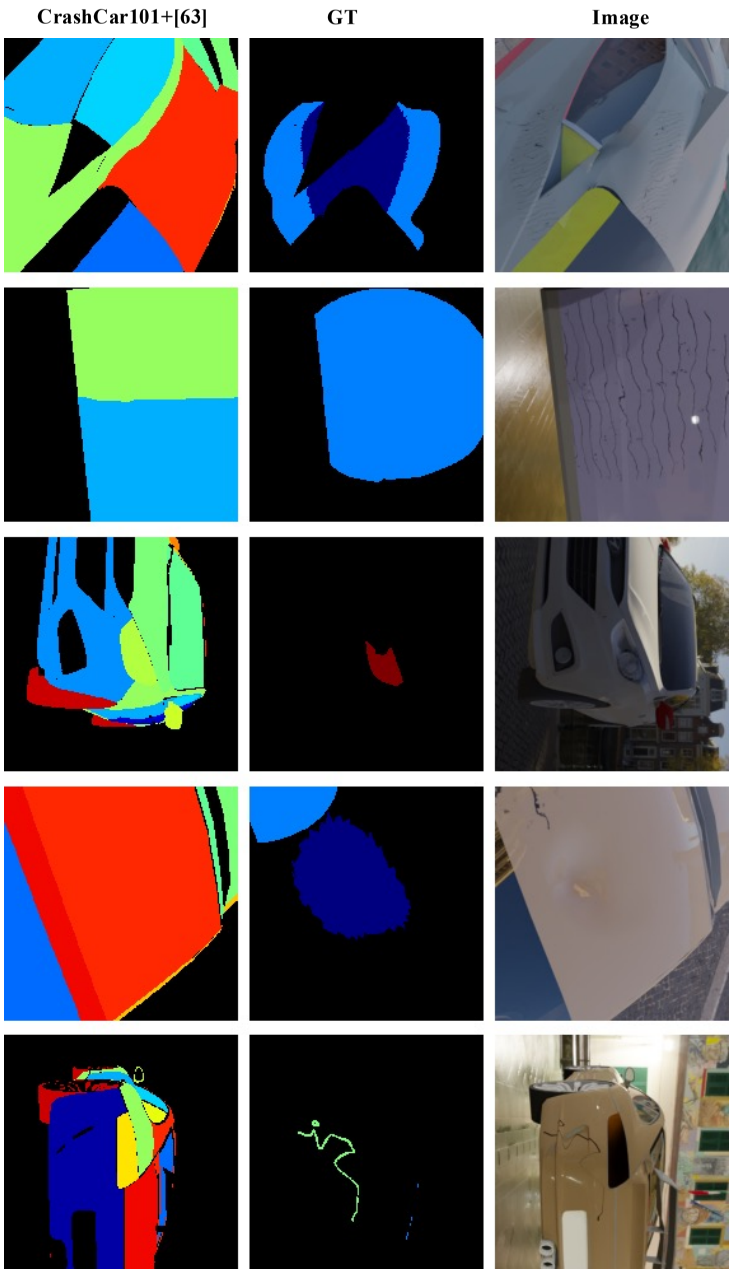

Parts
Damage
Image


Parts
Damage
Image


CrashCar101+[63]
GT
Image


CrashCar101+[63]
GT
Image


CrashCar101+[63]
GT
Image


CrashCar101+[63]
GT
Image