Vehicle Test and Validation,
Vehicle Operation

3D Vehicle Model (point cloud) processing in Python environment

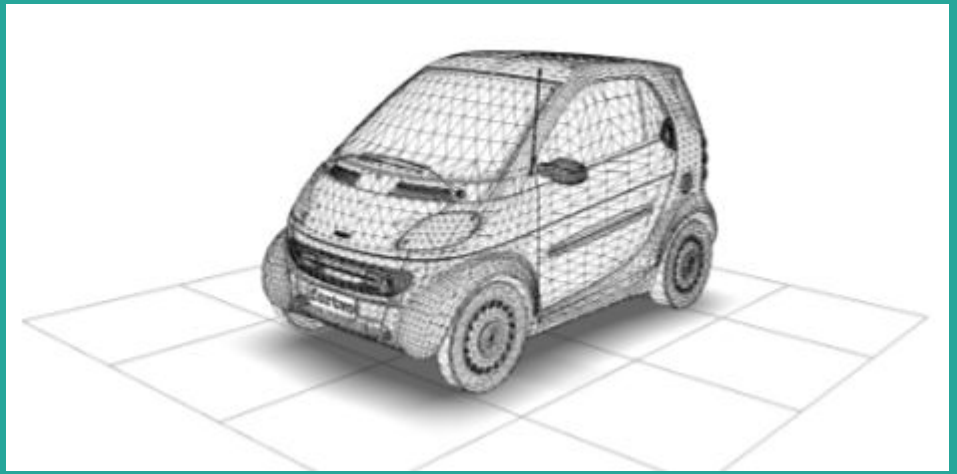
—

Igor Racca

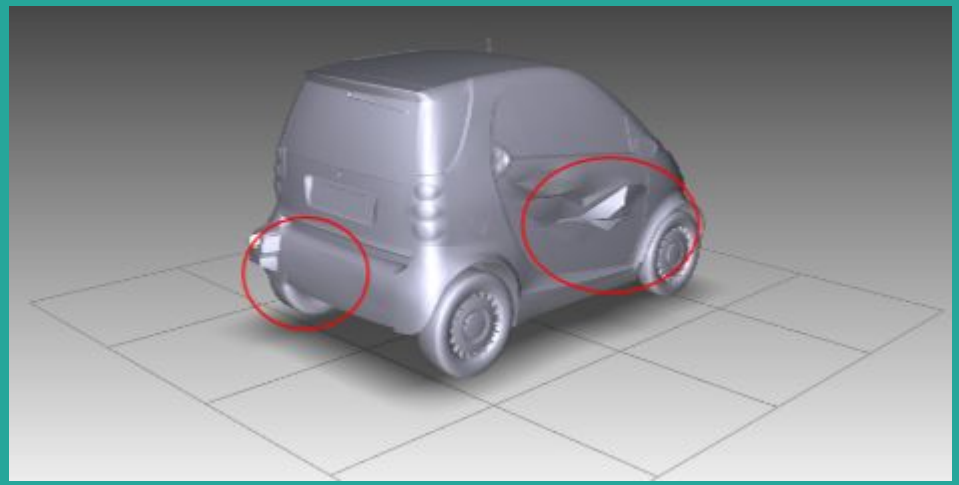
K45DZH

Project

- **Supervisor: Gábor Vida**
- **András Rövid**
- **Júlia Nagy**
- **Another student**



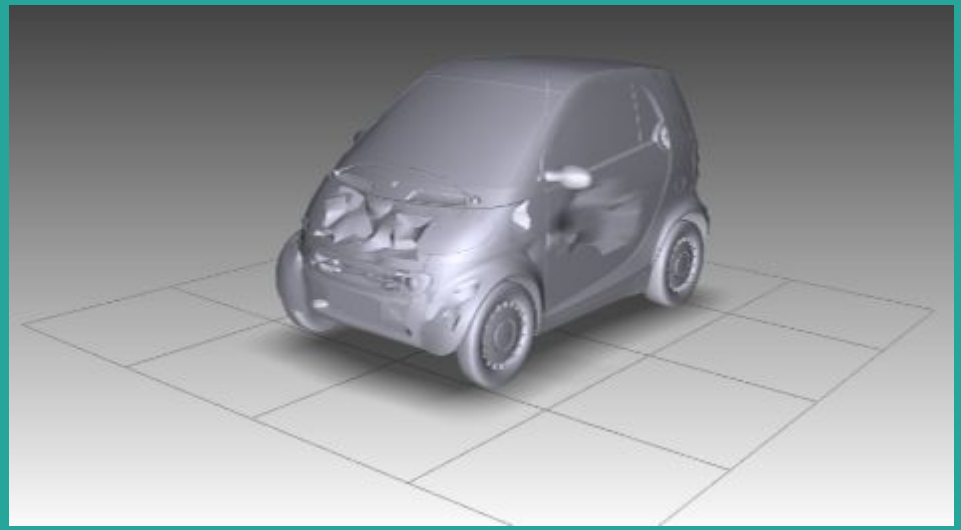
Motivation



Evaluate the amount of damage on a vehicle (if any)

Project

- **Neural Network**
- **Python, Pytorch**
- **Input - Vehicle Model**
- **Output - Damage**

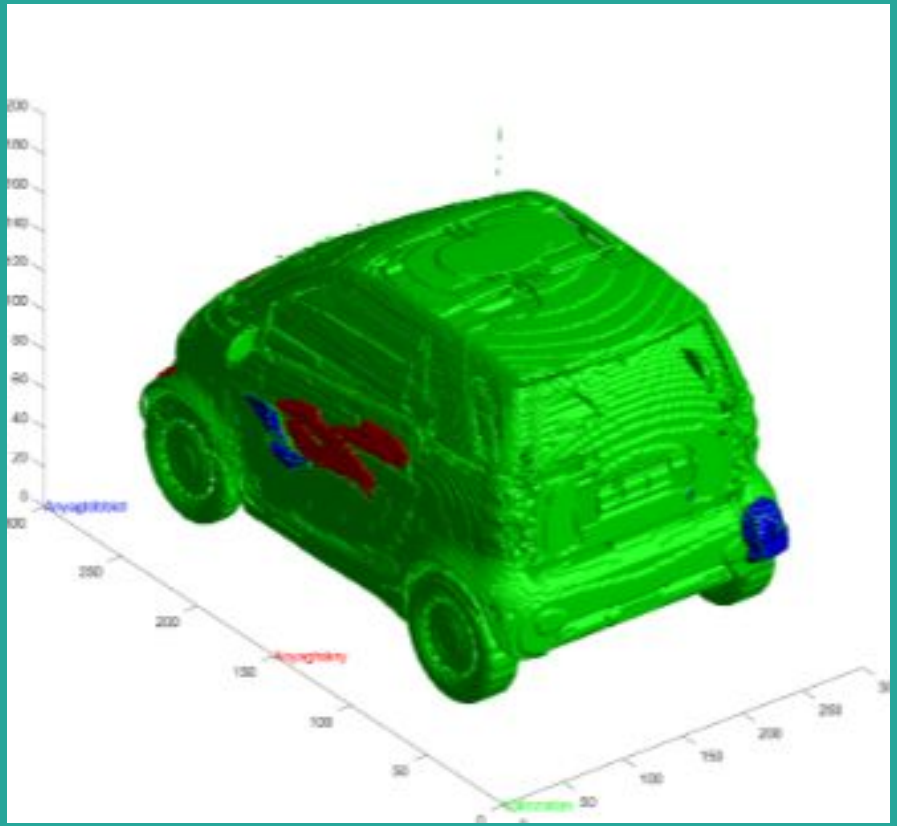


Input

- **Point Cloud**
- **Values:**

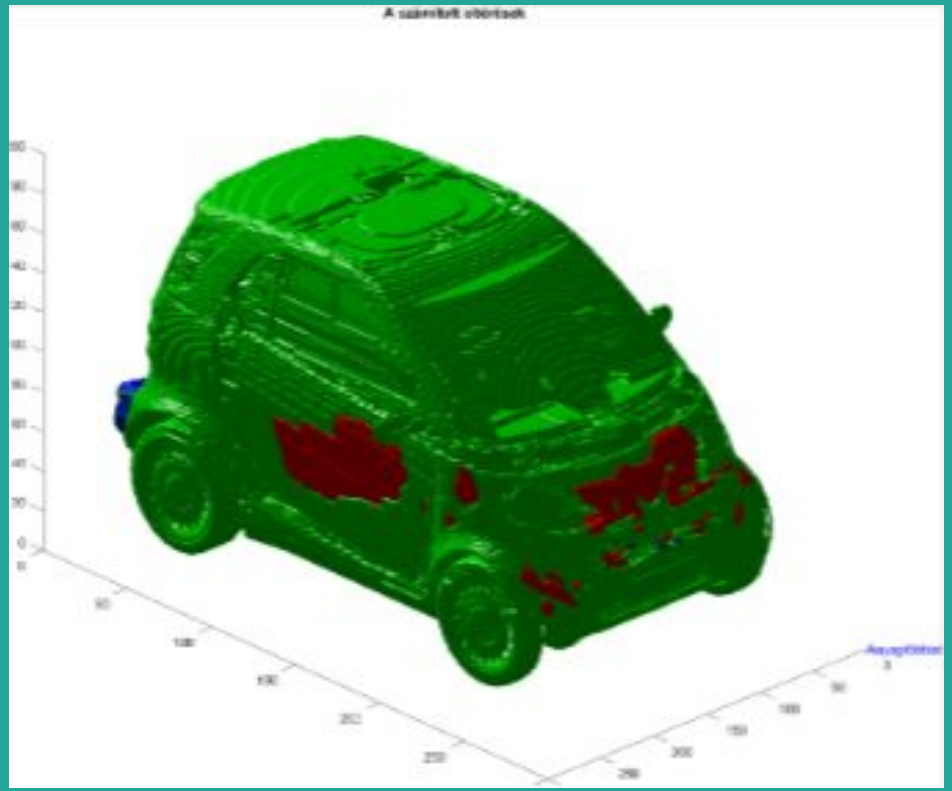
-1	Lack of volume
1	Extra volume
0	Normal volume

- **torch.Tensor**



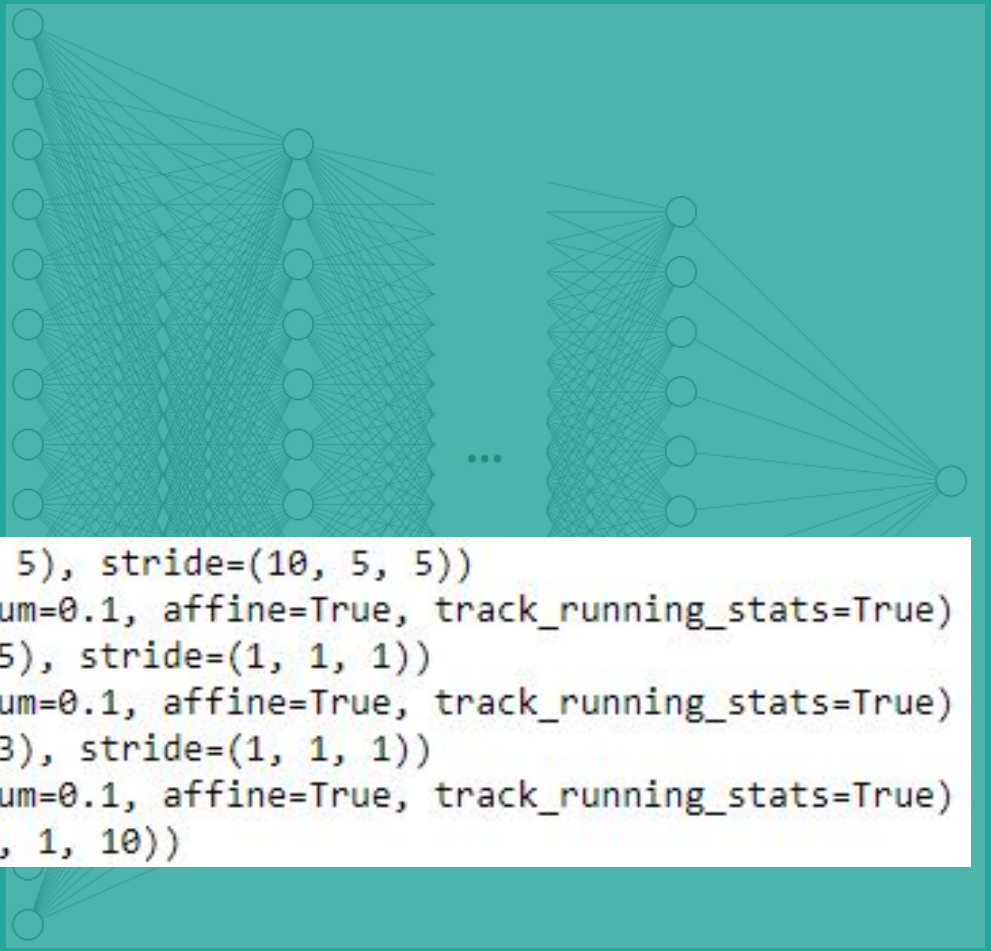
Input

- **Size: 500 * 200 * 200 = 20 M**
- **Vehicle models: 50**
- **Future changes**
  - **Inner points**
  - **New input: shell**
  - **Computational time**



NN - Model

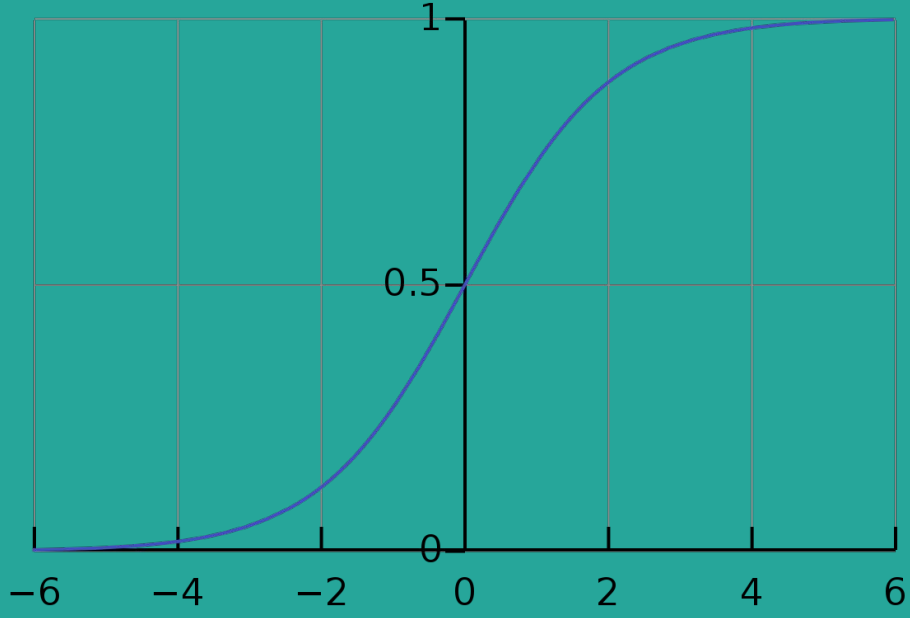
- **Model was previously made**
- **It was not changed**



```
(0): Conv3d(1, 3, kernel_size=(10, 5, 5), stride=(10, 5, 5))
(1): BatchNorm3d(3, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): Conv3d(3, 2, kernel_size=(5, 5, 5), stride=(1, 1, 1))
(3): BatchNorm3d(2, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(4): Conv3d(2, 1, kernel_size=(3, 3, 3), stride=(1, 1, 1))
(5): BatchNorm3d(1, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(6): AdaptiveMaxPool3d(output_size=(1, 1, 10))
```

NN

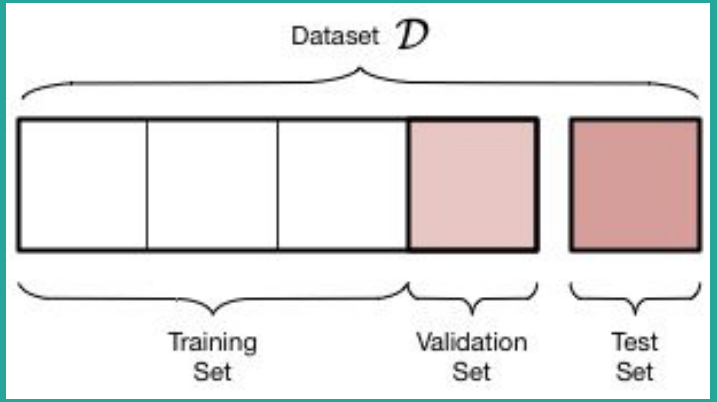
- **Loss function: MSE**
- **Output in [0,1]**
- **Convenient for percentages**
- **Activation function: Sigmoid**



NN - Dataset

- **Dataset: only 50 vehicle models**

- **Split Samples**
  - **Training: 30**
  - **Validation: 10**
  - **Test: 10**



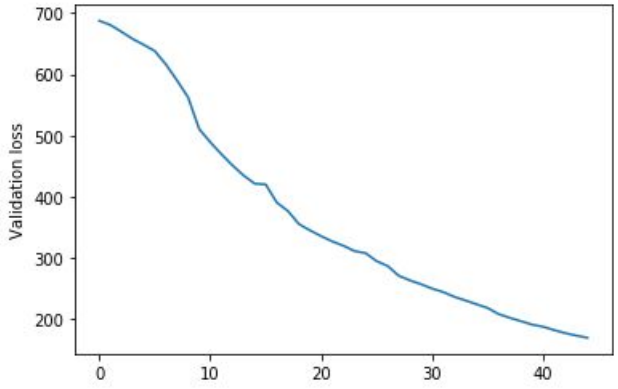
NN - Training and Validation

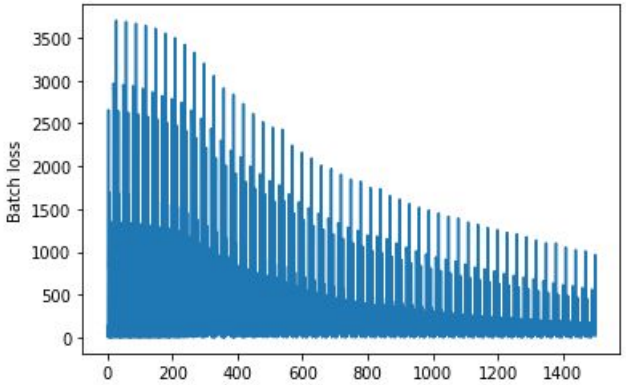
- **Batchmode (Batch size is equal to the total dataset)**
- **Shuffling training set (performed bad)**
- **Epochs = 50 is a good number**
- **Changing Learning Rate from 0.0001 to 0.01 resulted on a big improvement**

NN - Training and Validation

```
100%|██████████| 50/50 [01:15<00:00,  1.50s/it]
Training acc: 70.96774193548387 %
Validation acc:  60.0 %
saving weights is complete
```

$\mathrm{LR} = 0.0001$

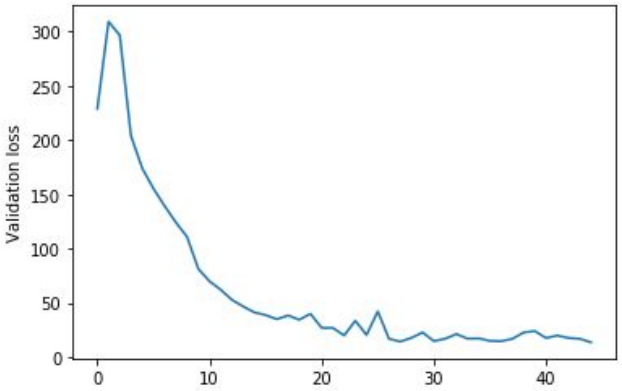


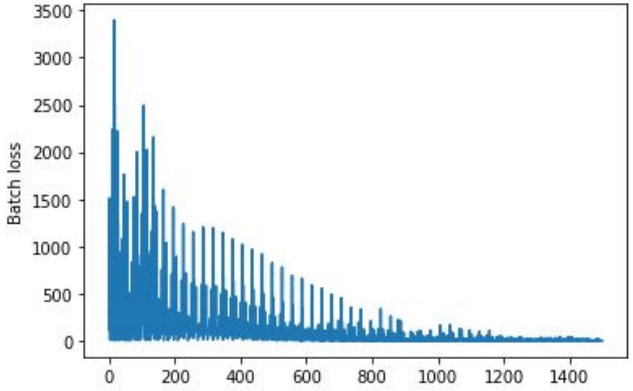


NN - Training and Validation

```
100%|██████████| 50/50 [01:16<00:00,  1.54s/it]
Training acc: 93.54838709677419 %
Validation acc:  90.0 %
saving weights is complete
```

$\mathrm{LR} = 0.01$



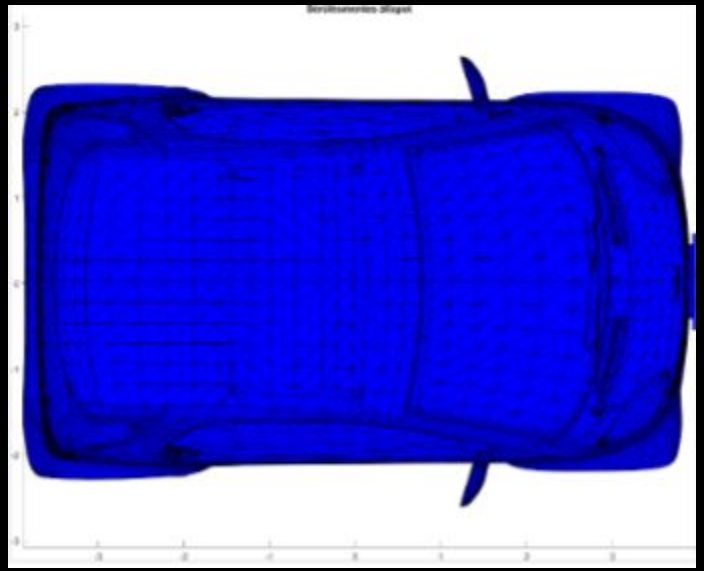


Improvements

- **Bad at predicting not damaged vehicles**
- **Try Mini-batches and shuffling**
- **Try Dropout**
- **Check Overfitting / Underfitting (more data needed)**

Conclusion

- **More data is required to better evaluation**
- **Optimization of the inputs is required to process more data**
- **This works will serve as documentation for next students**



Thank you !