Crack Detection from High-Quality Surface Images

David Malmgren-Hansen
Head of R&D, Greenwood Engineering

RPUG
Road Profile Users' Group

Project

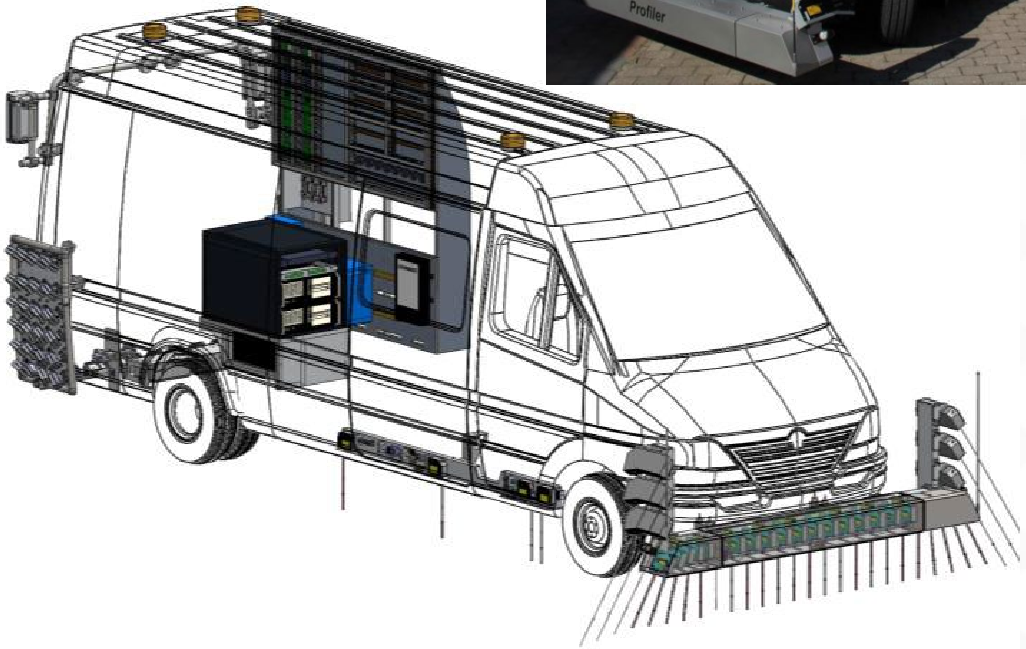
- The advances of Deep Learning algorithms makes them interesting for complex data analysis tasks to reduce manual labor.
- Can we use DL for Crack Detection?
- Problem: DL requires large amount of training data (images + labels)
- Vast amount of data has already been analyzed manually. Can this be used for DL training?
- Convolutional Neural Networks (CNN), a branch of DL is preferred in Image Analysis, among other disciplines.

Surface Imaging System

System specifications example

- 2 Linescan cameras + 64 LED lamps
- 4 m measuring width
- 1mm x 1mm per pix.
- Pulsed light system, 1 pulse/mm, <30kHz
- One continuous image of the surface
- Driving speed <100 km/h
- Red light source

  -> To be independent of sunlight
- Homogenous light profile



Crack Detection

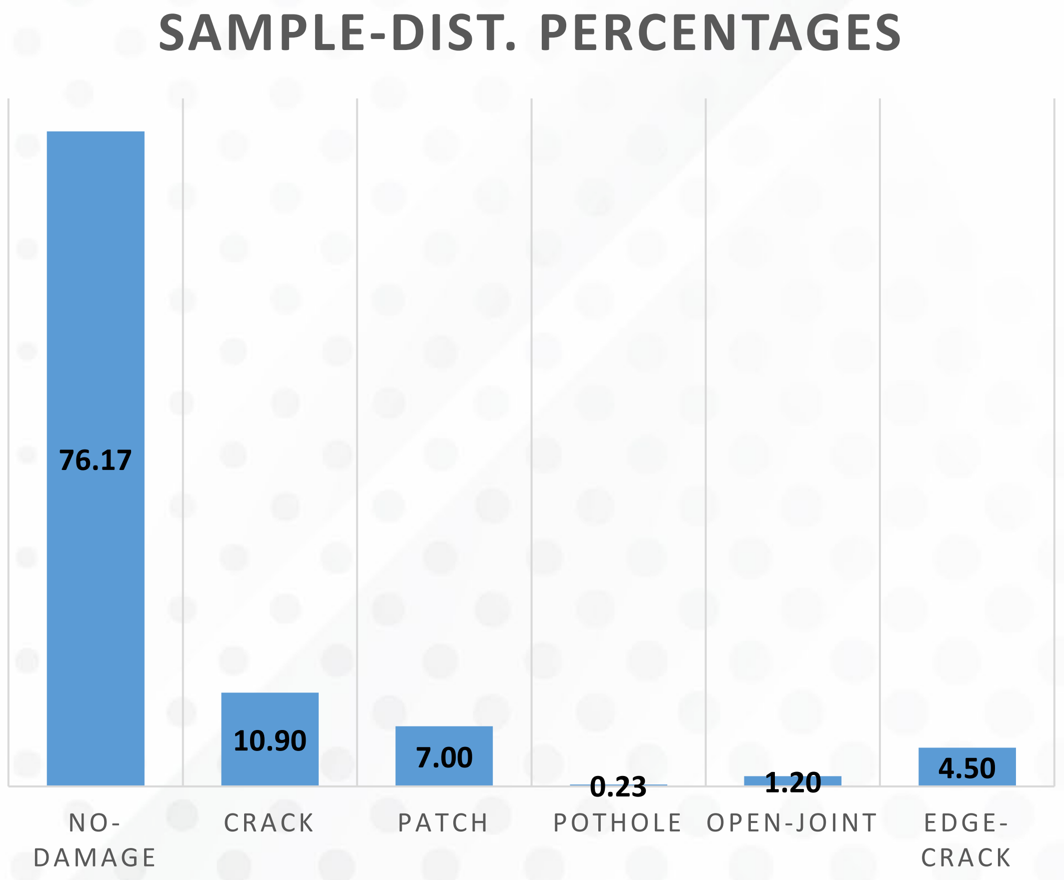
Greenwood Block Crack-Dataset

- 1.5M Images – 1358x991 pixels or appr. 1.3m x 1m, ~500 km of lane measurements.
- Blocks: 3 splits across **x** 1 row every ~1m
- Labels - ***Crack, Patch, Pothole, Open-joint, Edge-crack***
- Fraction of images where double checked
- Multi-label classification (non-exclusive classes) , i.e. several damage types in the same grid cell.
- All visible cracks are marked, down to ~1mm.

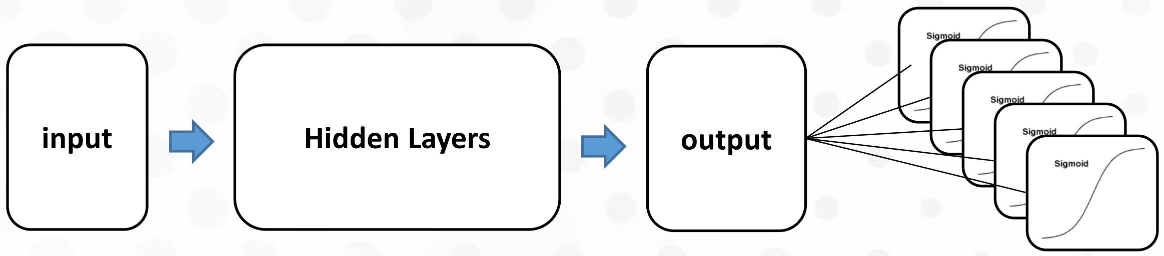
Crack Detection

Greenwood Block Crack-Dataset

- 1.5M Images – 1358x991 pixels or appr. 1.3m x 1m, ~500 km of road measurements.
- Blocks: 3 splits across **x** 1 row every ~1m
- Labels - ***Crack, Patch, Pothole, Open-joint, Edge-crack***
- Fraction of images where double checked
- Multi-label classification (non-exclusive classes) , i.e. several damage types in the same grid cell.
- All visible cracks are marked, down to ~1mm.



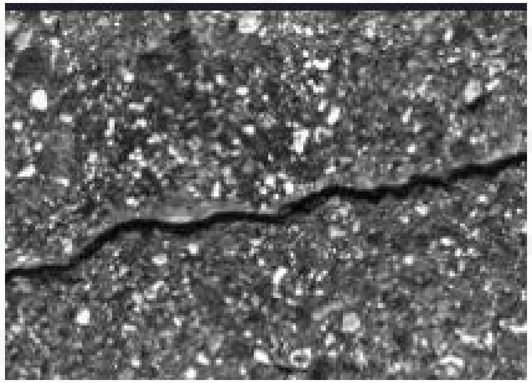
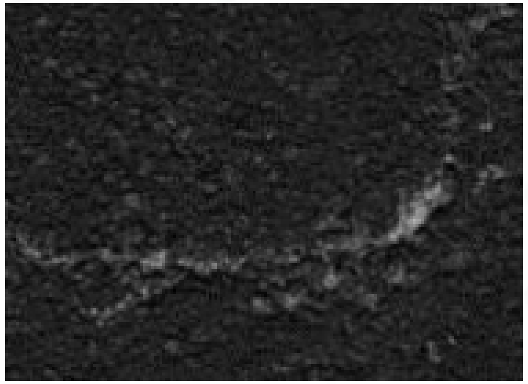
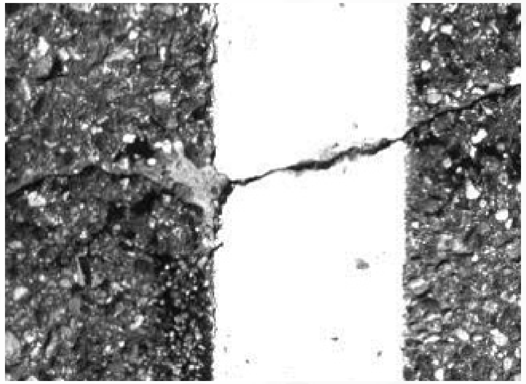
Classification types



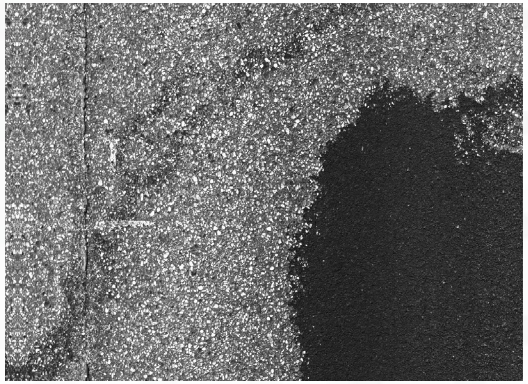
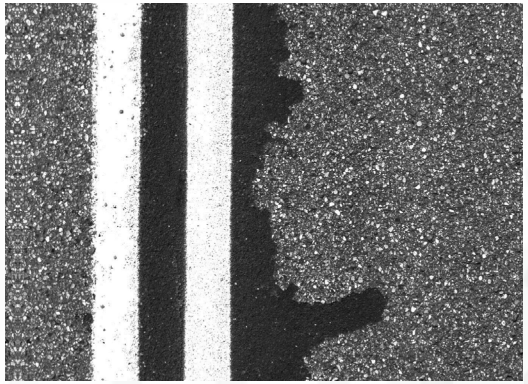
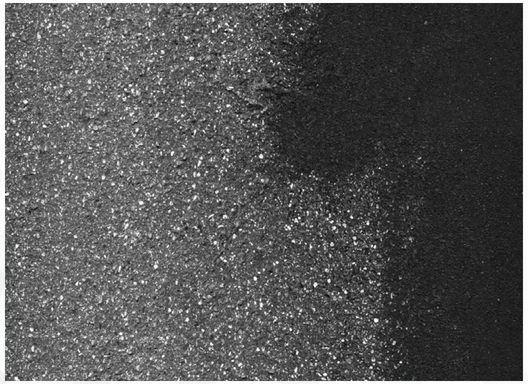
- Binary classification $\quad Sigmoid: \quad S(x) = \frac{e^x}{e^x+1}$
- Multiclass classification -> Mutual Exclusive $\quad Softmax: \quad S(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum_K e^x}, for\ K\ classes$
- **Multi-label classification**-> Non-exclusive $\quad Sigmoid: \quad S(x_i) = \frac{e^{x_i}}{e^{x_i+1}}$, for i = 1 … K classes

Dataset examples

Cracks

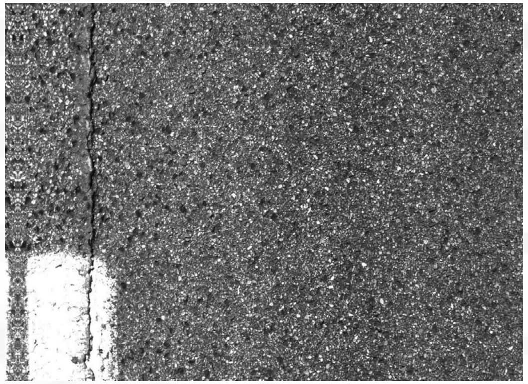




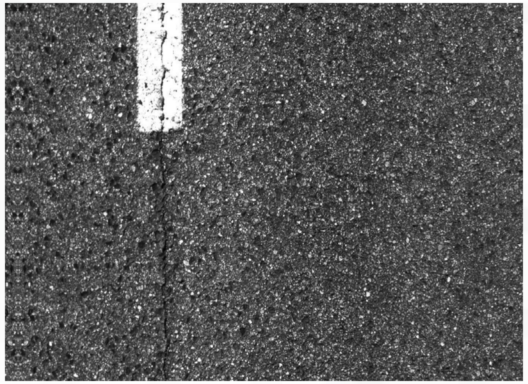
Patch

Dataset examples

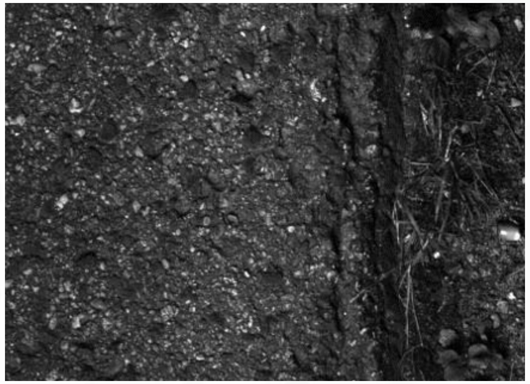
Open-joint

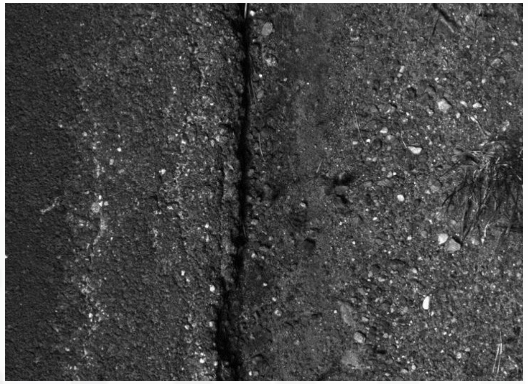




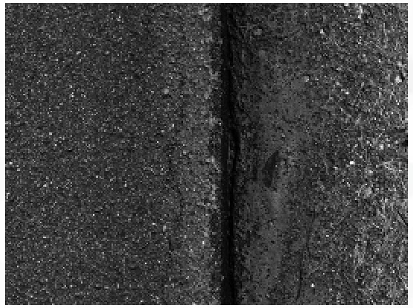
Potholes

Edge-crack

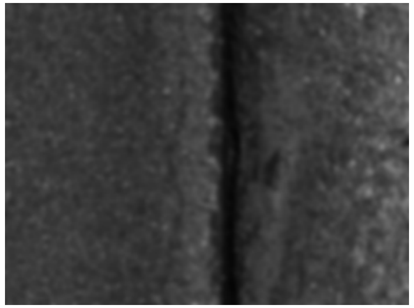




Convolutional Neural Network



$$* \quad \begin{array}{|c|c|c|} \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array} \quad =$$



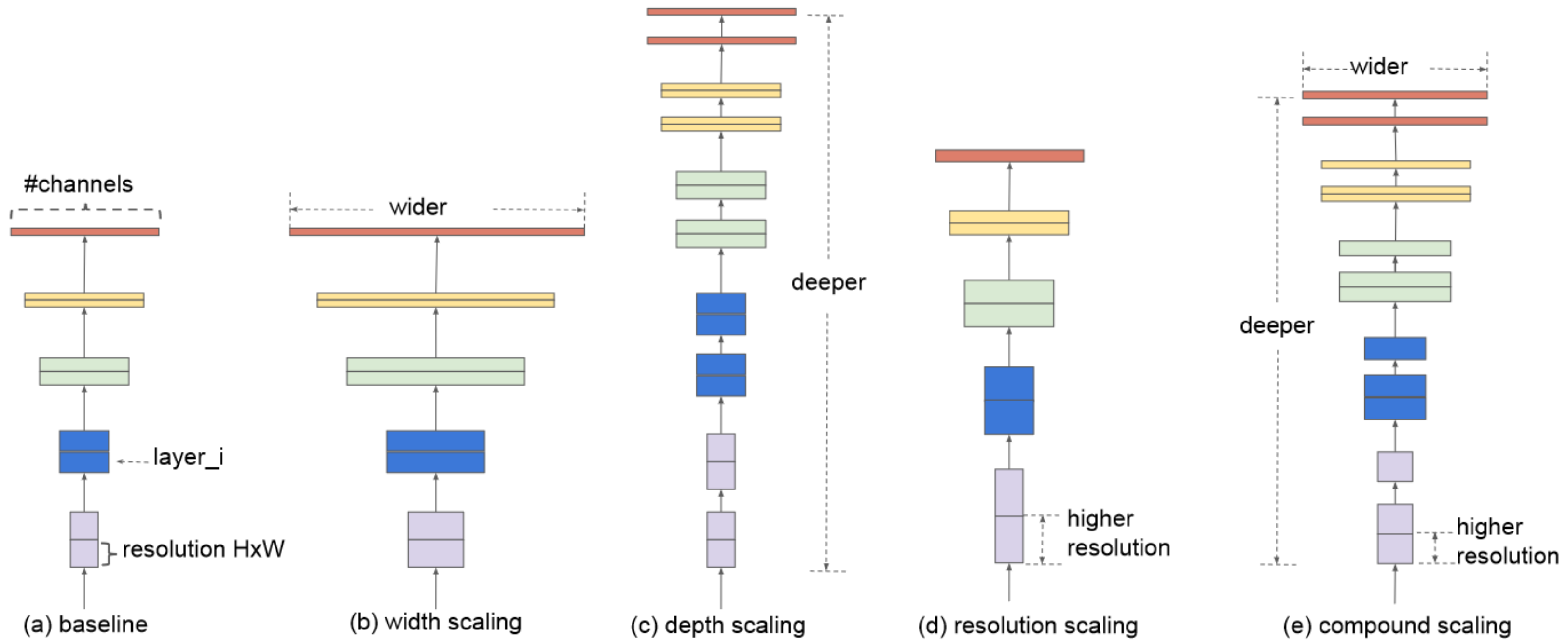
Convolution is a filtering operation (e.g. blur above)

Modern networks has

- thousands filters
- stacked in layers (< 100-200), i.e. filtered images are parsed into new filters.
- This yields hundred of millions parameters

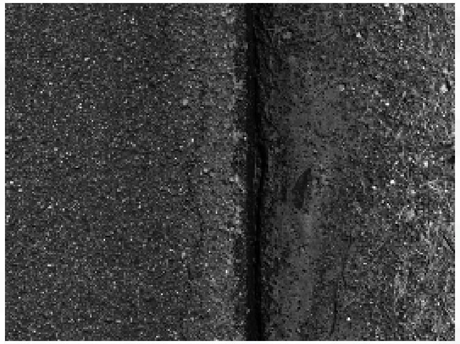
Although regularization is used, a lot of data is needed.

CNN Architectures

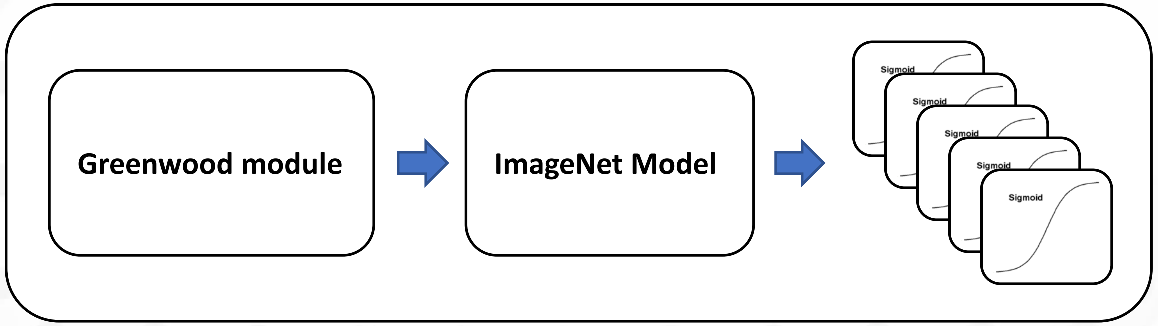


ImageNet models:
DenseNet, Xception, MobileNet, InceptionResNet, NasNet, **EfficientNet**

Customized model



1358x991 pixels

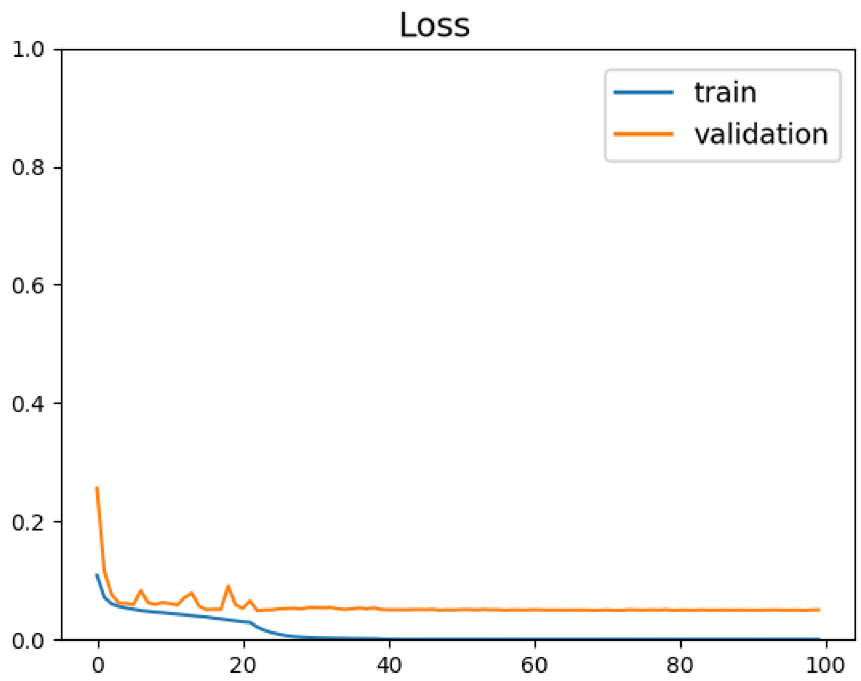


Crack = 88%
Patch = 5%
Pothole = 0%
Open-joint 0%
Edge-crack = 15%

- The Greenwood Module consist of a convolutional layer and a subsampling layer.
- This is to reduce the large amount of computations needed in the ImageNet model which normally takes images of 300x300 pixels.
- Typical ImageNet models takes weeks to train on a non-cluster setup on ~1M images.

Training

- Training time 3-15 days on fast PC (i9 CPU + 2xGPU-RTX3090)
- When model is trained it can process 420 km/h
- Oversampling under sampled classes
- Class weights -> put more “attention” to underrepresented classes
- ADAM optimizer + Learning rate decay



Results - Accuracies

Accuracies below is calculated as,

$$\boldsymbol{Acc} = \frac{TP + TN}{2}$$

This is important since No-damage is overrepresented, which is often the case in pavement surfaces.

Potholes must be oversampled more, possibly with data-augmentation.

Balancing Strategy *Labels*	Oversampling	Oversampling + Class-weights
crack	**90,1**	89,5
patch	**93,4**	92,2
pothole	56,8	**58,6**
open joint	81,9	**84,1**
edge crack	**89,5**	87,9
Overall	**90,5**	89,4

Results

- Co-occurrence matrix shows columns-wise percentages of a damage type occurring with the other damage types
- This means one damage tends to lead to other damages as well.

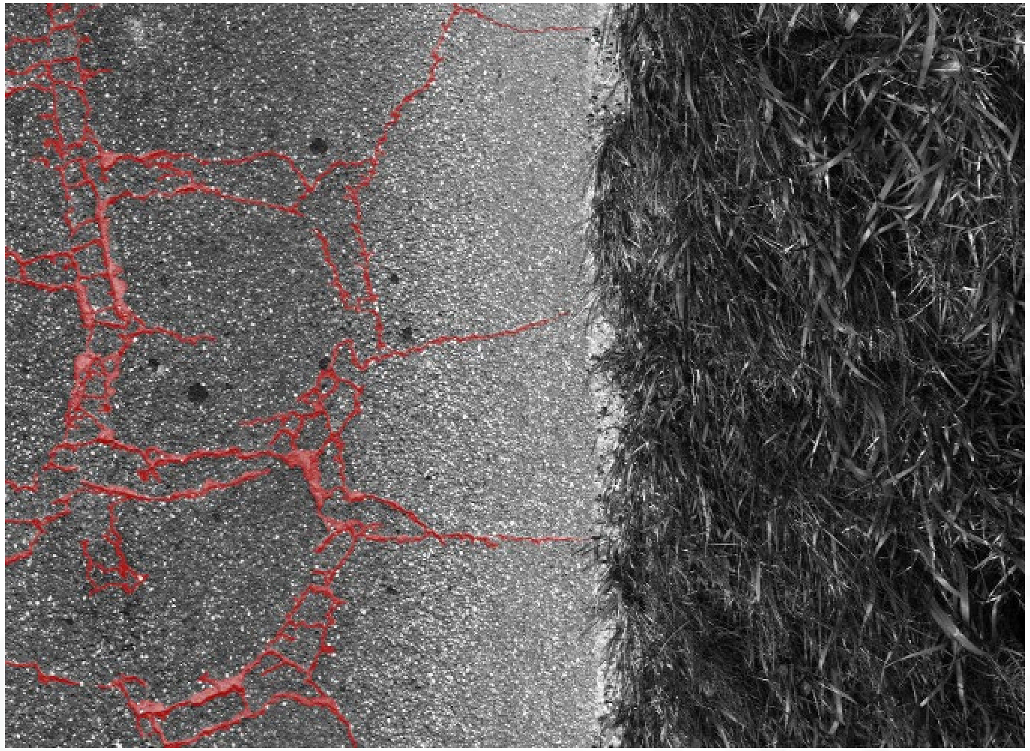
Co-occurrence matrix

	crack	*patch*	*pothole*	*open_joint*	*edge_crack*
crack	100,0	48,5	67,5	22,9	95,8
patch	33,2	100,0	58,2	11,7	31,4
pothole	1,3	1,6	100,0	0,3	1,4
open joint	2,8	2,1	1,7	100,0	0,3
edge crack	40,4	19,3	31,7	0,9	100,0

- When we overall categorize a block for a damage, we capture most damages.

crack	*patch*	*pothole*	*open_joint*	*edge_crack*
89,1	93,7	89,7	78,8	90,9

Next step



Greenwood Crack Segmentation Dataset

Fine grained information:

- Pct damage per cell
- Width (max, mean std)
- Orientation (Longitudinal/Transversal/Both)
- Monitor individual cracks over time

Conclusion

- Deep Learning achieves high accuracy on detection of surface damages.
- Challenge: Highly imbalanced data
- Oversampling and weight decay can help, but..
- … more samples are needed for underrepresented classes.
- As overall accuracies are high, and FP are low the model is very good for screening large networks and find AOIs.
- Adding more classes is easier with a model to prescreen AOI.