Generalization capabilities and performance analysis of CNNs for pavement crack detection

Tianyu Mo
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
t.mo@student.utwente.nl

ABSTRACT

Frequent loading may cause pavement cracks, which have hidden dangers to vehicles and pedestrians. Additionally, the cracks will also deteriorate over time thus the cracks should be found out in time. The current crack detection method is to recognize them by naked eyes, but it is time-consuming and non-accurate. Hence, it is necessary to develop automatic crack detection models. Currently, CNN-based road crack detection methods get more popular. The author evaluates the performance, generalization capabilities, and image processing time of state-of-the-art methods on three datasets using ODS, OIS, AP, processing time metrics. And the experiment results show the HED performs better in these three aspects, which reaches above ODS 0.75 in some datasets. The possible reason why HED is better at performance and generalization in crack detection is it outperforms on thin crack identification and robustness. Additionally, it also shows a greater prospect of real-time crack detection.

Keywords

Crack detection, Generalization capability, Convolutional neural network

1. INTRODUCTION

Cracks are a common road defect, which can increase with time and pose a threat to the safety of drivers. So it is important to find out and repair cracks to reduce the maintenance fees and ensure safety before they continue to deteriorate[10, 22]. Currently, detecting cracks manually is a relatively common way. However, the detection process was considered as being time-consuming and the detection results seem to be subjective. Hence, automatic crack assessment is crucial for the future pavement crack detection[9].

Currently, there are many different emerging crack detection approaches based on different mechanisms like 3D data, or deep learning[2].

3D data allows the crack to have spatial structure. The model based on 3D data uses one more dimension to depict the cracks and it means there is more information about cracks can be used for the crack analysis such as the depth, the volume of a crack, which is helpful to enhance the robustness of the system[2]. However, even though 3D data can provide more complete information about the cracks, it also increases the complexity of the algorithm in the meantime. Currently, deep convolutional neural networks contribute to solving the high computational cost issue in models based on 3D data[2].

Deep learning-based models are popular in the crack detection field. When compared to traditional machine learning techniques, feature extraction in deep learning can be done automatically and the model based on deep learning also has a better performance on generalization than the one based on traditional machine learning due to bigger data usage[2].

CNN-based models showed high precision on computer vision-related tasks[13]. And generalization capabilities play an important role in the resistance of minor image transformations and distortions[14]. Since the trained models typically are used on different roads and should be able to recognize cracks accurately, it has high requirements for generalization capabilities and performance, otherwise, the model could only be able to handle some specific cases and can not replace manual detection. Hence, an evaluation of state-of-the-art models seems crucial and this problem leads to two research questions:

Question 1 What is the best-performing CNN for pavement crack detection in the state-of-the-art?

Question 2 Which CNN generalizes better on different data sets in the state-of-the-art?

The paper aims to address the issue in the pavement crack detection to enable readers to get insights into the difference of generalization capabilities and performance of CNN-based models.

The rest of the paper is organized in the way below:

The "Related research" section introduces popular state-of-the-art CNN-based methods and the main evaluation metrics used in crack detection. The "Datasets" section describes the datasets used for the training and evaluation. The "Research method" section discusses the metrics for the evaluation. The "Measurement environment" section describes the experiment environment and parameter setting. The "Results and discussion" section presents and analyses the experiment results. The "Conclusion" section summarizes the work conducted.

2. RELATED RESEARCH

Arbeláez et al.[1] used ODS, OIS, AP to evaluate method performance in contour detection. By comparing the metric results of methods, it is able to clearly see the performance difference between different methods. Yang et al.[18] proposed a way to evaluate generalization capacity by using means of ODS and OIS, which can show whether the model performs stably in different datasets.

Some CNN-based methods perform well in crack detec-

34th Twente Student Conference on IT Jan. 29th, 2021, Enschede, The Netherlands.

tion. Xie et al.[16] proposed an edge detection method called HED (holistically-nested edge detection) to solve edge detection problems. There were many edge detection methods developed based on different principles. However, with the rapid development of deep learning, CNN-based edge detection methods show better performance. The holistically-nested edge detection method is able to accept the image as the input and output the result in the image directly, which is what "holistically" means here. In addition, "nested" represents the current layer can inherit the results of previous ones so that the model can have better performance on consistency, otherwise there is a higher probability of spatial shift issue. After solving issues in holistic images multi-scale and multi-level feature learning well, HED is considered to be good at accuracy and computational speed.
Ronneberger et al.[15] proposed an image segmentation method based on convolutional networks called U-Net to solve object recognition challenges. Convolutional networks show good performance on the image segmentation issues, but the size of the dataset is an obstacle to further improvement. After introducing ImageNet[4], this issue seems to be solved. However, it also means that the large dataset gets to a necessity to train a high-performance model, but it is not always possible to obtain it in different fields. Ciresan et al.[3] tried to solve this issue by using the sliding-window technique. But there are two main issues here. The first one is slowness, since it is necessary to run the network for each patch in this case. And the other one is the high requirements on the number of max-pooling layers, which could decrease the accuracy. To solve these issues, U-Net uses "fully convolutional network"[11] architecture and it is added more feature channels in the upsampling part in the meanwhile, which causes the expansive path to seem symmetric to the contracting one and also makes the architecture look like "u". Besides, U-Net uses elastic deformations to expand the datasets so that it can perform well on small datasets when compared to other methods.

3. DATASETS

The CRACK500[19, 20], GAPs384[6, 19], Cracktree200[21] datasets are used for the evaluation, which contains 3348, 509, 206 raw images with the dimensions of 2560 to 1440, 640 to 540, 800 to 600 pixels respectively. For each raw image, there is a corresponding groundtruth. In addition, there are 1124, 39, 21 images are used for the test in each dataset mentioned above. And the rest of the images will be divided up into the training set and validation set at a ratio of 9:1.

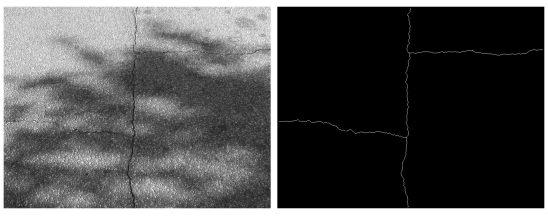

Figure 1. A crack image and the corresponding groundtruth image in Cracktree200 dataset

Cracktree200: Zou et al.[21] introduced a dataset of pavement crack images that has 206 images with the dimension of 800 * 600 pixels and Yang et al.[18] named it as Cracktree200, which involves pavement cracks images with different recognition challenges like shadows, thin cracks like images shown in Figure 1.

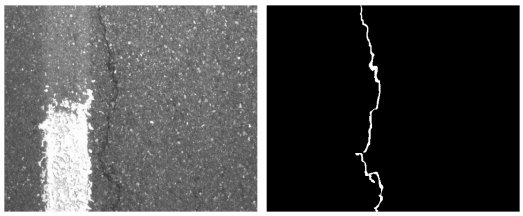

Figure 2. A crack image and the corresponding groundtruth image in GAPs384 dataset

GAPs384: Eisenbach et al.[7] presented GAPs dataset which contains 1969 images with dimensions of 1920 * 1080 pixels. Yang et al.[18] chose 384 images from it to build a dataset called GAPs384. They also cropped each image into 6 parts to fit the GPU limitation and there are 509 images in the dataset eventually. There are different challenges with the crack detection like white lines on the road, low contrast, sealed cracks, and Figure 2 shows an image and corresponding groundtruth in the GAPs384 dataset.

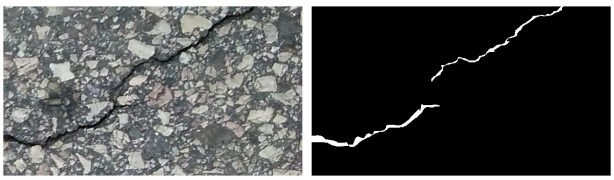

Figure 3. A crack image and the corresponding groundtruth image in CRACK500 dataset

CRACK500: Yang et al.[18] presented CRACK500 dataset, which consists of 500 2000 * 1500 pixel images of pavement cracks in Temple University. In addition, it is divided into three parts for training, validation, testing, which contains 250, 50, 200 images respectively. To get more images, the images are cropped into 12 parts and the dataset has 1896, 328, 1124 images for each function mentioned above. Figure 3 shows an example in the CRACK500 dataset and the spaces between road grit will be one of the crack detection obstacles for the methods.

4. RESEARCH METHOD

The state-of-the-art crack detection models e.g. HED[16, 17], U-Net[15, 12] are chosen for testing and comparing their generalization capabilities and performance.
There are three metrics that are used for the performance and generalization evaluation: ODS, OIS, AP. Due to the similarity to edge detection, the metrics for edge detection can be used for the evaluation. ODS is short for optimal dataset scale, which represents the best F-measure in different dataset scales. And OIS means optimal image scale, which is the aggregate F-measure in the best scale for each image. The last one is AP, which represents the average precision here[5, 18].

$$ODS = max\{2\frac{P_t \times R_t}{P_t + R_t} : t = 0.01, 0.02, ..., 0.99\} \quad (1)$$

$$OIS = \frac{1}{N_{img}} \sum_{i}^{N_{img}} max\{2\frac{P_t^i \times R_t^i}{P_t^i + R_t^i} : t = 0.01, 0.02, ..., 0.99\} \quad (2)$$

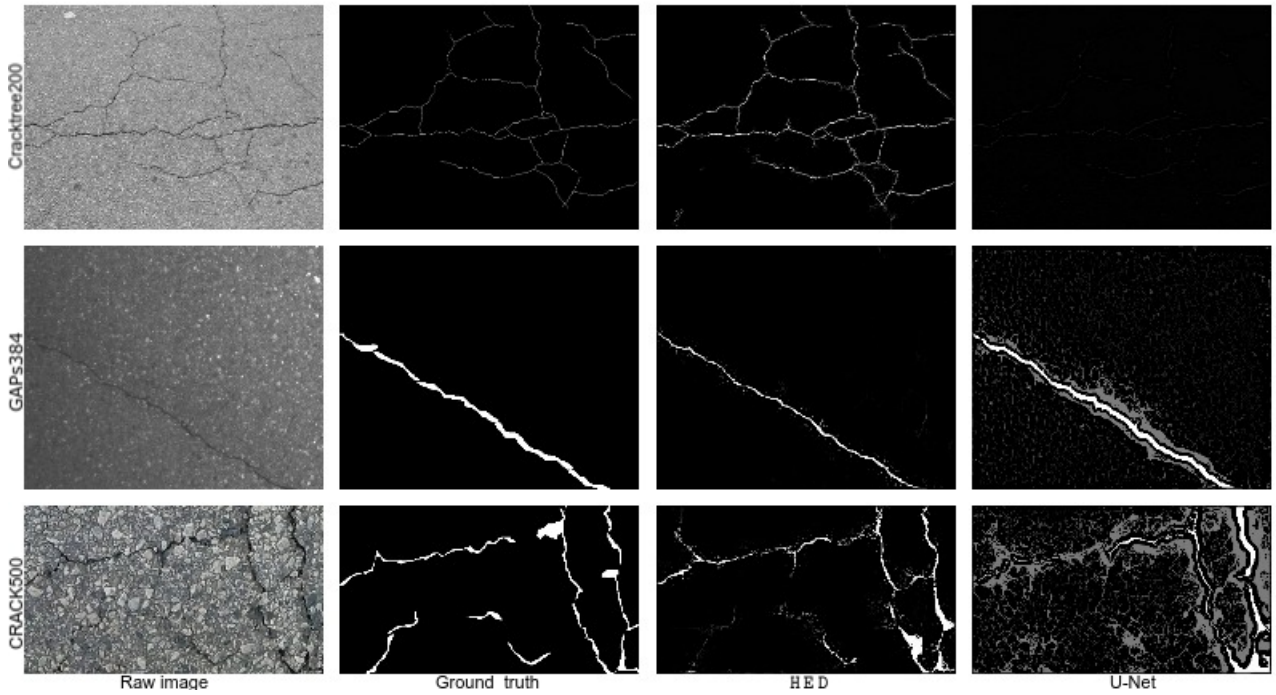

Figure 4. Visualization of method performance on different datasets

where:

P	Precision
R	Recall
t	Threshold
i	Index of image
N_{img}	Number of images

To evaluate the performance of each method, calculate the ODS, OIS, AP values of methods on each dataset. And compare the results of these three metrics between methods to find out the best-performing one.

To evaluate the generalization capabilities of methods, train the models with the first dataset, and then test the trained models on the rest of the datasets successively. And the means and standard deviations of ODS, OIS, AP are used for the evaluation, which can be calculated by using other dataset results of methods trained on a specific dataset. For instance, for HED trained on the Cracktree200, the means and standard deviations can be calculated by using the ODS, OIS, AP results of this model on GAPs384, CRACK500. And also compare the mean and standard deviation results between methods to choose one with the best generalization capability.

5. MEASUREMENT ENVIRONMENT

5.1 Platform

All training processes were conducted on Google Colab with 12GB NVIDIA Tesla K80 GPU. And the evaluation part was performed on the laptop with Intel dual-core i5 CPU @2.7GHz by using the tool from [8].

5.2 Parameter Setting

All methods are trained based on SGD optimizer and following hyperparameters: batch size(1), step size(1e4), momentum(0.9), weight decay(2e-4), gamma(0.1)

The initial learning rate is 1e-3, which is multiplying by gamma(0.1) every step size(1e4) iterations to find out a model with the best performance.

6. RESULTS AND DISCUSSION

6.1 Performance

Cracktree200 Dataset: As shown in the Table 3. We can see HED has better performance on Cracktree200 than U-Net. HED trained on Cracktree200 can reach 0.925, 0.953, 0.818 in ODS, OIS, AP respectively. But the model of U-Net trained on the same dataset only achieve 0.469, 0.477, 0.430 in ODS, OIS, AP from the second row of Table 3.

GAPs384 Dataset: According to results in the Table 4. HED outperforms U-Net slightly, which has a similar score on the ODS. However, HED has better performance on OIS and AP than U-net, which are 0.004, 0.112 higher.

CRACK500 Dataset: From results in the Table 5, we can see HED is better on all three metrics than U-Net. Specifically, HED outperforms U-Net by 16.6%, 23.3%,30.0% in ODS, OIS, AP respectively.

In general, the HED method has better performance than U-Net in crack detection. Since the HED achieves better results on three datasets in the performance evaluation. A possible reason is it has better performance on thin crack identification and robustness. For instance, from the 4, the result of HED on CRACK500 shows more continuity on the thin cracks when compared to the result of U-Net. For the robustness, the result of HED on GAPs384 shows a cleaner background, but U-Net probably also takes particles in the raw image into account.

6.2 Generalization Capability

Cracktree200 Dataset: According to results in the Table 6, HED trained on Cracktree200 has better generalization capability than U-Net. HED achieves better results on means of ODS, OIS, AP. In detail, they are 0.256, 0.245, 0.184 respectively.

	HED trained on Cracktree200			HED trained on GAPs384			HED trained on CRACK500		
Datasets	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
Cracktree200	0.925	0.953	0.818	0.776	0.807	0.829	0.379	0.437	0.262
GAPs384	0.267	0.262	0.210	0.266	0.235	0.193	0.286	0.301	0.189
CRACK500	0.244	0.229	0.158	0.351	0.292	0.290	0.498	0.560	0.385

Table 1. Evaluation results of HED trained on Cracktree200, GAPs384, CRACK500 respectively

	U-Net trained on Cracktree200			U-Net trained on GAPs384			U-Net trained on CRACK500		
Datasets	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
Cracktree200	0.469	0.477	0.430	0.709	0.736	0.090	0.147	0.219	0.068
GAPs384	0.105	0.088	0.039	0.271	0.191	0.081	0.166	0.171	0.086
CRACK500	0.032	0.023	0.005	0.143	0.141	0.011	0.427	0.454	0.296

Table 2. Evaluation results of U-Net trained on Cracktree200, GAPs384, CRACK500 respectively

Method	ODS	OIS	AP
HED	0.925	0.953	0.818
U-Net	0.469	0.477	0.43

Table 3. Evaluation results of method trained on Cracktree200

Method	ODS	OIS	AP
HED	0.266	0.235	0.193
U-Net	0.271	0.191	0.081

Table 4. Evaluation results of method trained on GAPs384

Method	ODS	OIS	AP
HED	0.498	0.56	0.385
U-Net	0.427	0.454	0.296

Table 5. Evaluation results of method trained on CRACK500

GAPs384 Dataset: As shown in the Table 7, HED and U-Net both show decent generalization capability. But HED still has higher scores on the means of all the three metrics than U-Net. And the AP means of HED is 0.503 higher than U-Net's.

CRACK500 Dataset: From results in the Table 8, HED shows better generalization capability than U-Net. And HED's means of metrics are 0.176, 0,174, 0.149 in ODS, OIS, AP higher than U-Net's respectively.

Method	ODS	OIS	AP
HED	0.256±0.012	0.245±0.017	0.184±0.026
U-Net	0.069±0.037	0.056±0.033	0.022±0.017

Table 6. The mean and standard deviation of ODS, OIS, and AP over datasets GAPs384, CRACK500 of methods trained Cracktree200

Method	ODS	OIS	AP
HED	0.563+0.213	0.550+0.258	0.560+0.270
U-Net	0.426+0.283	0.439+0.298	0.051+0.040

Table 7. The mean and standard deviation of ODS, OIS, and AP over datasets Cracktree200, CRACK500 of methods trained GAPs384

Method	ODS	OIS	AP
HED	0.333+0.047	0.369+0.068	0.226+0.037
U-Net	0.157+0.010	0.195+0.024	0.077+0.068

Table 8. The mean and standard deviation of ODS, OIS, and AP over datasets Cracktree200, GAPs384 of methods trained CRACK500

Generally, the HED method also performs better than U-Net on generalization capability in crack detection due to better stability in three datasets in generalization capability evaluation. A possible reason is the same as the one in the performance part: the HED is better at the thin-crack identification and robustness than U-Net. For instance, from the Figure 6, HED trained on GAPs384 shows a more accurate result than U-Net on Cracktree200 since the width of cracked recognized by HED is closer to the one in the ground truth. For the robustness, the HED trained on GAPs384 shows fewer errors than U-Net, because some spaces between grits are falsely recognized as cracks and there are also more noises caused by grits in the background of the result of U-Net. Besides, From the Figure 5 and 7, the results of HED also show fewer noises caused by grits in the backgrounds than the ones of U-Net.

6.3 Processing Time

Method	Cracktree200	GAPs384	CRACK500
HED	0.135s	0.080s	0.083s
U-Net	0.272s	0.333s	0.485s

Table 9. The time the method takes to process an image in specific dataset on GPU

As shown in Table 9, HED processes images faster than U-Net on the three datasets, which takes 0.099s on average. It seems like HED shows the greater potential on the real-time image processing in crack detection according to the test results. And if there is a computer with a proper GPU on board of a car that checks the roads, it could be a better method choice for real-time crack detection.

7. CONCLUSION

CNN-based methods play an increasingly important role in road crack detection. In this paper, the author tried to find out the CNN-based methods with the best performance and with best generalization capability, and with

the fastest image processing speed by testing them on different pavement crack datasets. The result shows the HED is better at performance, generalization capability, and processing time than U-Net in crack detection. The possible reason why it achieves better results on performance and generalization in crack detection is this method is good at thin-crack identification and not easy to be disturbed by background noises. Besides, it also has a greater potential for real-time crack detection in the meanwhile.

8. REFERENCES

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[2] W. Cao, Q. Liu, and Z. He. Review of pavement defect detection methods. *IEEE Access*, 8:14531–14544, 2020.

[3] D. Cireşan, A. Giusti, L. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, page 2843–2851, Red Hook, NY, USA, 2012. Curran Associates Inc.

[4] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] P. Dollár and C. Zitnick. Fast edge detection using structured forests. *CoRR*, abs/1406.5549, 2014.

[6] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H. Gross. How to get pavement distress detection ready for deep learning? a systematic approach. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2039–2047, 2017.

[7] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H. Gross. How to get pavement distress detection ready for deep learning? a systematic approach. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2039–2047, 2017.

[8] Fyangneil. Pavement crack detection: dataset and model. *GitHub*.

[9] J. Huyan, W. Li, S. Tighe, Z. Xu, and J. Zhai. Cracku-net: A novel deep convolutional neural network for pixelwise pavement crack detection. *Structural Control Health Monitoring*, 27, 2020.

[10] Z. Li, G. Xu, Y. Cheng, Z. Wang, and Q. Wu. Pavement crack detection using progressive curvilinear structure anisotropy filtering and adaptive graph-cuts. *IEEE Access*, 8:65020–65034, 2020.

[11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation, 2015.

[12] Milesial. Unet: Semantic segmentation with pytorch. *GitHub*.

[13] Ç. Özgenel and A. Gönenç Sorguç. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 35, pages 1–8. IAARC Publications, 2018.

[14] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, and X. Shen. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials*, 234:117367, 2020.

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[16] S. Xie and Z. Tu. Holistically-nested edge detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403, 2015.

[17] W. Xu. A pytorch reimplementation of hed. *GitHub*.

[18] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling. Feature pyramid and hierarchical boosting network for pavement crack detection, 2019.

[19] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling. Feature pyramid and hierarchical boosting network for pavement crack detection, 2019.

[20] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3708–3712, 2016.

[21] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[22] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3):1498–1512, 2019.

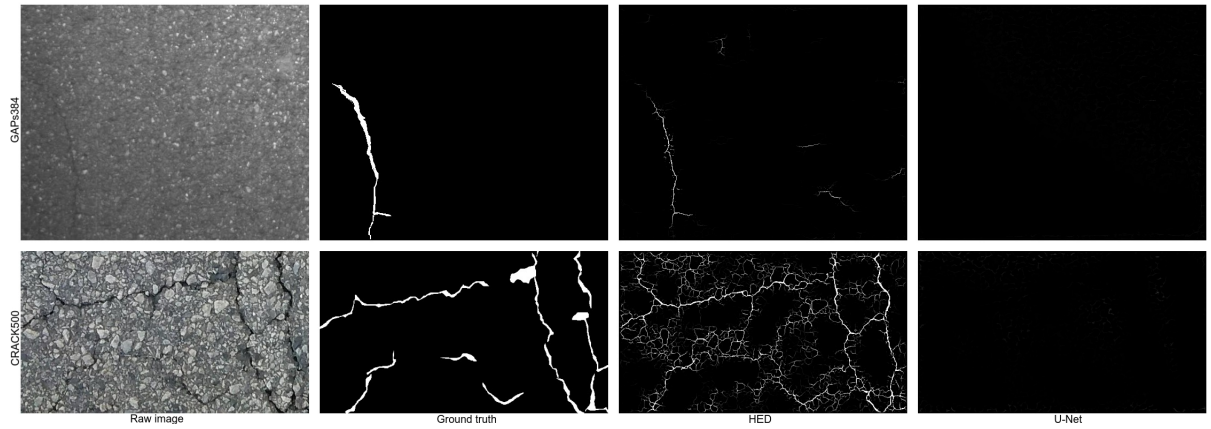

Figure 5. Visualization of generalization capabilities of methods trained on Cracktree200 over datasets GAPs384, CRACK500

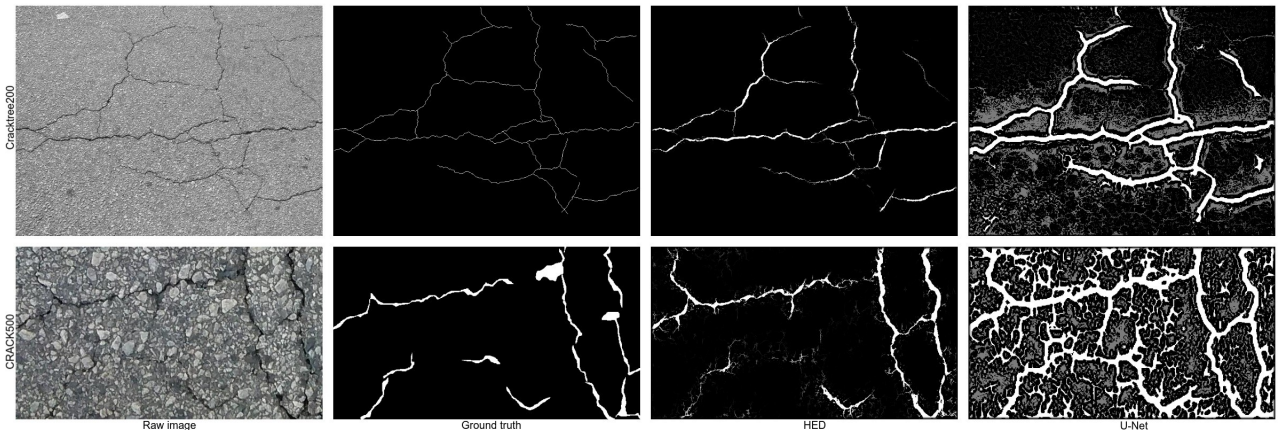

Figure 6. Visualization of generalization capabilities of methods trained on GAPs384 over datasets Cracktree200, CRACK500

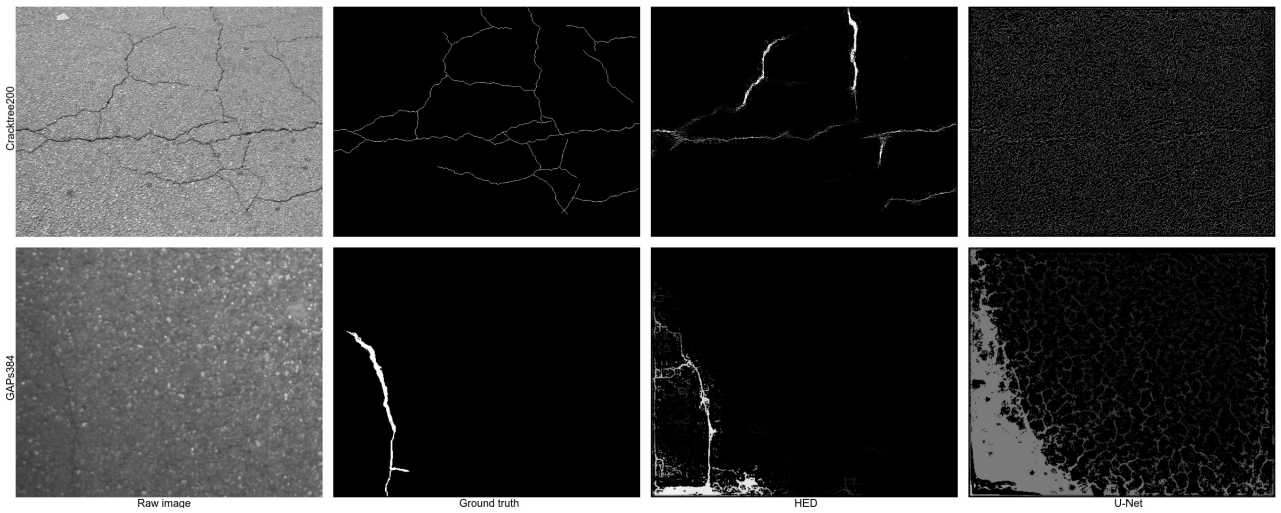

Figure 7. Visualization of generalization capabilities of methods trained on CRACK500 over datasets Cracktree200, GAPs384