ITM Web of Conferences **59**, 02013 (2024) https://doi.org/10.1051/itmconf/20245902013
HMMOCS-II 2023

Methods for detecting and counting nodes in images of crack networks

Alexey Rybakov*

Astrakhan State University, Astrakhan, Russian Federation

Abstract. The article discusses a technique for segmenting a network of cracks in micrographs and identifying the main elements such as a node, the junction of several cracks, and an edge, the body of the crack itself, to build a model of the network as an undirected graph. Crack segmentation was carried out using two methods: using threshold binarization and applying masks that separate nodes from edges based on morphological characteristics, and a combined method using a convolutional neural network to detect nodes. Such methods make it possible to detect nodes and edges automatically, facilitating the construction of a model and opening up new possibilities in theoretical calculations of the resistance of a network of conductors in transparent conductive coatings.

1 Introduction

Currently, there is great interest in technologies for creating transparent conductive coatings. Transparent conductive films can be used for large area transparent heaters critical to display panels, touch screens, solar panels, camera lenses, windows and mirrors that require high conductivity in cold and humid environments. Transparent heaters are inevitable where protection from fogging and icing is required for stable operation of the devices. [1]

Moreover, transparent conductive films can be used for shielding from electromagnetic radiation. Metal micro- and nanostructures are most suitable for such protection due to their low surface resistance and high optical transmittance [2,3].

For the successful use of such coatings, not only practical methods for their manufacture are of great importance, but also theoretical methods for modeling the resistance of the network of conductors that make them up. This paper proposes segmentation methods using computer vision algorithms for images of a network of conductors obtained on microcrack patterns. A number of authors also pay great attention to the problem of crack detection and segmentation [4,5]. These studies are mainly related to the construction industry, in particular the detection of cracks on the walls of buildings or other structures in order to predict their possible further destruction, however, they provide a good starting point for creating a technique for segmenting images of microcracks. Li et al. [6] proposed a crack segmentation algorithm based on threshold binarization. Schmugge et al. [7] used a deep learning method to separate image pixels into cracks and background, that is, they reduced the solution to a classical segmentation problem. However, the result showed relatively low accuracy. Nguyen

* Corresponding author: rybakov_alex@mail.ru

[8] divided a crack image into sub-images, then performed feature extraction to construct feature vectors and trained to identify the sub-images. Zhang [9] used deep learning method to detect cracks in block-based classification. However, the proposed method leads to inconvenient detection and is sensitive to block size. Walesa [10] used triangles to divide the image, high-pass filters to enhance the image, and binarized the image using the Otsu method. Subsequently, a morphological algorithm was used to synthesize a rectangular image, thereby detecting cracks by eliminating surface heterogeneity. Hu [11] used crack segmentation based on texture analysis. To separate cracks from the background, local binary pattern operators were used to determine whether each pixel belonged to a crack. However, this method does not take into account information about the local surroundings, resulting in the inability to accurately detect cracks with uneven intensity. McCormack [12] used the domain weighted averaging method to preprocess the gray image, and then combined the local threshold method and shape filtering to obtain a crack detection algorithm. Katsigiannis et al [13] proposed a deep learning approach based on transfer learning for crack detection in masonry facades. They specifically optimized the detection model to make it applicable to different types of brick walls. In [14], the authors addressed the limitations of FCN and Unet architectures by implementing spatial pyramid pooling (SPP) and advanced convolution modules in the DeepLabV3+ architecture. An analysis of the sources discussed above suggests that a combination of classical computer vision algorithms and neural network models will make it possible to solve the problems posed in the work.

2 Materials and methods

This article examines the possibility of segmenting cracks and identifying such objects as the crack body itself (edge) and the crack junction (node or vertex). As an example, microphotographs provided by colleagues from the Federal Research Center “Krasnoyarsk Scientific Center of the Siberian Branch of the Russian Academy of Sciences” were used. Cu–Ag and Ni–Ag networks were obtained by galvanic deposition of copper and nickel onto a thin Ag seed mesh produced by the cracking matrix method.[2]. To train the neural network model, photographs of cracks taken from open sources were used. The Roboflow tool [15] was chosen as a program for marking images. To implement computer vision methods, the Python programming language and the OpenCV library were used [16]. YOLO v8 [17] was used as a neural network model.

Next, the detected nodes and edges can be compared with each other and presented in the form of a graph, which will allow us to consider the network of cracks as a mathematical model for the purpose of calculating, for example, the resistance between individual sections on it.

3 Technique for segmenting images of cracks and determining their constituent elements

Threshold image binarization was proposed as an initial algorithm. To do this, the image is first converted to grayscale, then threshold levels are set and all pixels whose brightness is above the threshold level are assigned the maximum brightness value. The remaining pixels are assigned the minimum value. These values were chosen based on the 8-bit image representation as 255 and 0, respectively.

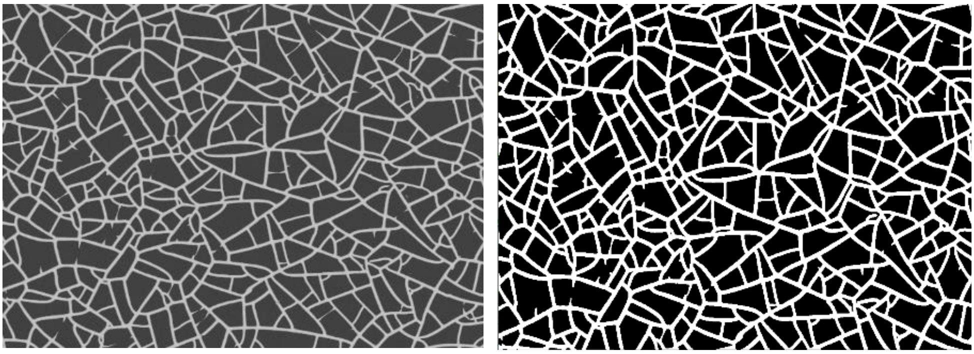

Fig. 1. Original (left) and binarized (right) images of the crack network.

Next, erosion and dilatation operations are performed to separate the nodes from the ribs (Figure 2). Dilatation (morphological expansion) – convolution of an image or a selected area of an image with a certain kernel. The core can have any shape and size. The kernel can be thought of as a template or mask. Applying dilation boils down to passing the template over the entire image and applying a local maximum search operator to the intensities of the image pixels that are covered by the template. Erosion (morphological narrowing) is the reverse operation. The effect of erosion is similar to dilatation, the only difference is that the local minimum search operator is used. [18]. A certain difficulty in this case is the selection of mask parameters, since the width of the cracks and the sizes of the nodes can vary significantly. This is one of the main disadvantages of such an algorithm.

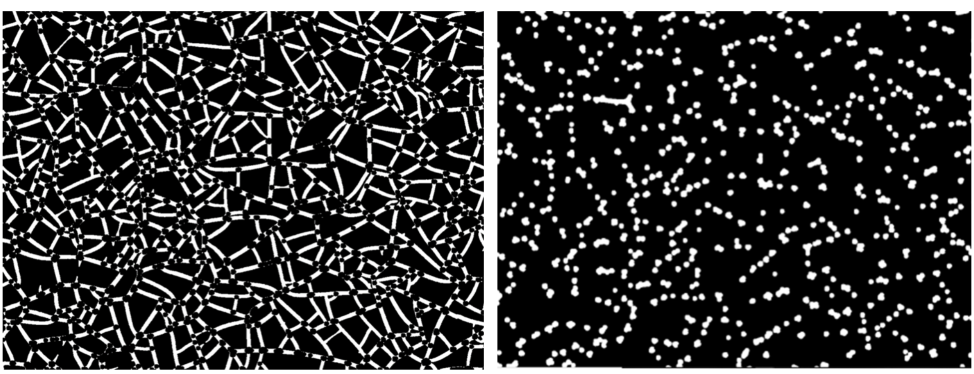

Fig. 2. Selecting nodes and edges in an image.

After selecting nodes and edges in the images, their contours are searched. The OpenCV library findContours() function extracts contours from a binary image using the algorithm [19].

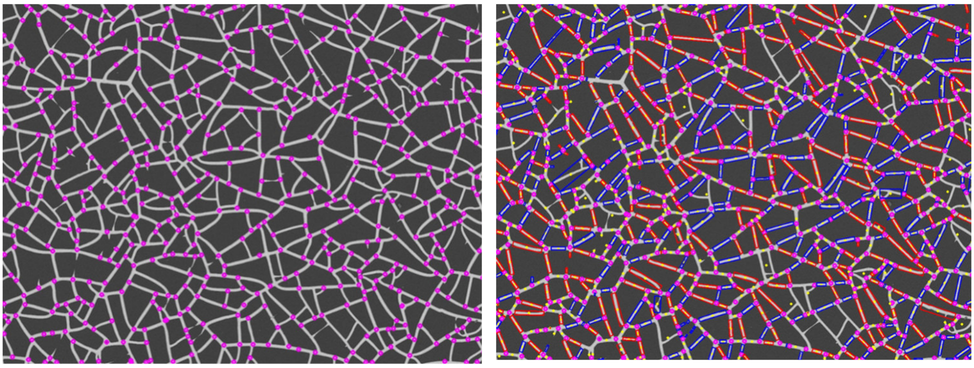

Fig. 3. Displaying node centers and edge boundaries in the source image.

Figure 3 shows the display of the found nodes and edge boundaries on the original image of the crack network. Blue rectangles display edges whose top end is to the right of the bottom, red rectangles - vice versa. This separation helps to obtain thin, elongated contours that better fit the crack model. At the same time, the aspect ratio of such rectangles can be used to construct a weighted graph.

As a result of the application of classical image processing algorithms, including binarization, morphological operations and search for contours, about 10-15% of nodes in the original images were not detected. This error can increase even more in the case of blurry, low-contrast, noisy images. To solve the problems of detecting the position of nodes on a network of cracks, the YOLO v8 convolutional neural network model was used. YOLOv8 is a modern neural network model that builds on the success of previous versions of YOLO and introduces new features and improvements to further improve performance and flexibility. YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking tasks, instance segmentation, image classification, and pose estimation [17].

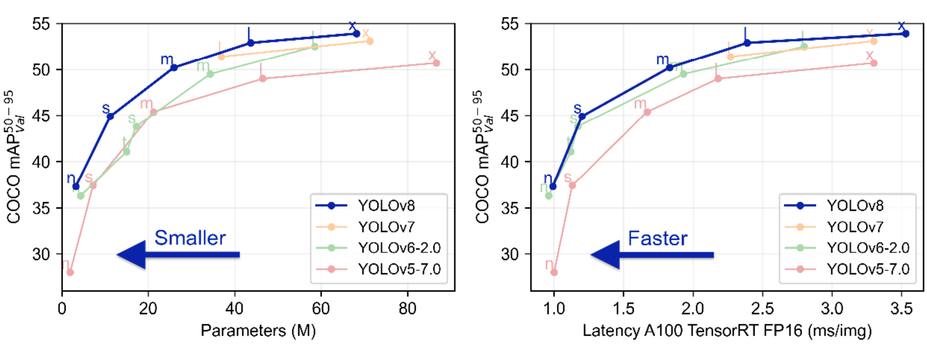

Fig. 4. Comparison of YOLOv8 version with previous ones.

204 images were selected as the initial dataset, divided in the following ratio: 182 training, 14 validation and 8 test images. The resolution of all images was changed to 640 by 640 pixels. Training took place over 250 epochs on an NVIDIA GeForce RTX 3060 GPU with different batch sizes and several optimizer options. The images were previously augmented using the rotation, reflection, and brightness tools.

4 Results and discussion

As a result of using only classical basic computer vision algorithms when detecting nodes and edges, a systematic error arose due to the inability to fine-tune the parameters of dilatation and erosion masks. This led to inaccuracies in matching nodes and edges. The comparison itself was carried out on the basis of searching for the closest neighboring edges to the vertex by limiting the maximum distance between them (Figure 5).

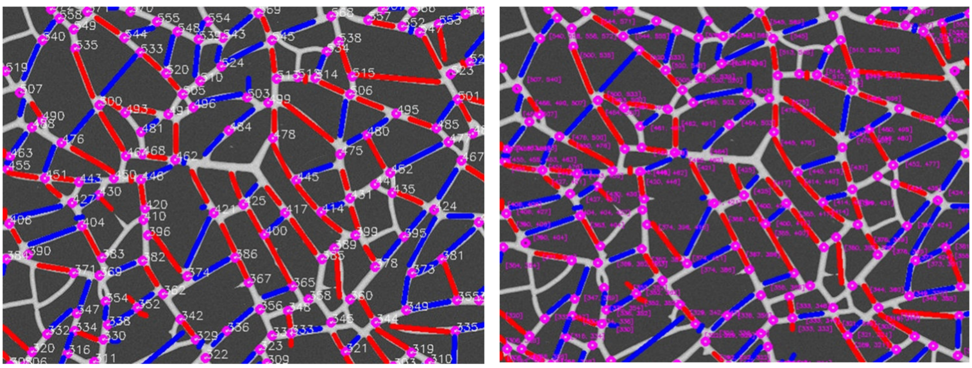

Fig. 5. Matching nodes and edges in an image.

The use of a neural network model made it possible to improve the accuracy of node detection, while eliminating the need to adjust the parameters of the morphological operations mask for each new image. For low-contrast, blurry images, this method turned out to be much more effective, allowing in some cases to detect more than 98% of all nodes automatically. The result of node detection is shown in Figure 6.

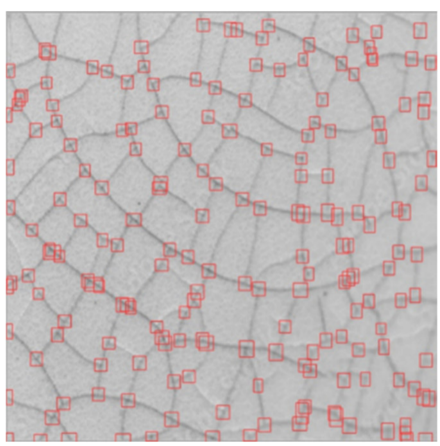

Fig. 6. Result of node recognition using a pretrained neural network

For the original image of the network of cracks, which we chose as a standard, it was necessary to perform an inversion, since mainly images of dark cracks on a light background were used as a training dataset.

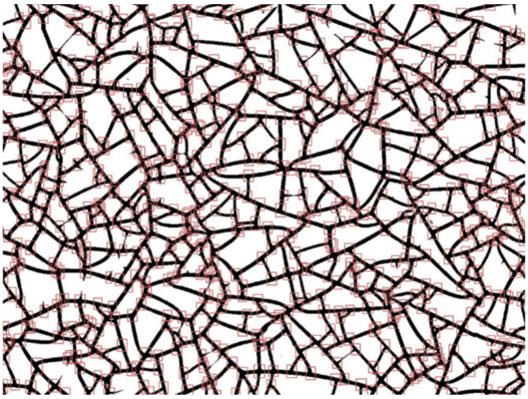

Fig. 7. Result of node recognition in the original image.

The study compares the experimental results obtained using such criteria as accuracy, precision, recall, and F1 score. The definitions of metrics characterizing the results of the model are described in [20]. As a result of training the model, an error matrix with TP (True Positive) values of about 0.87 was obtained.

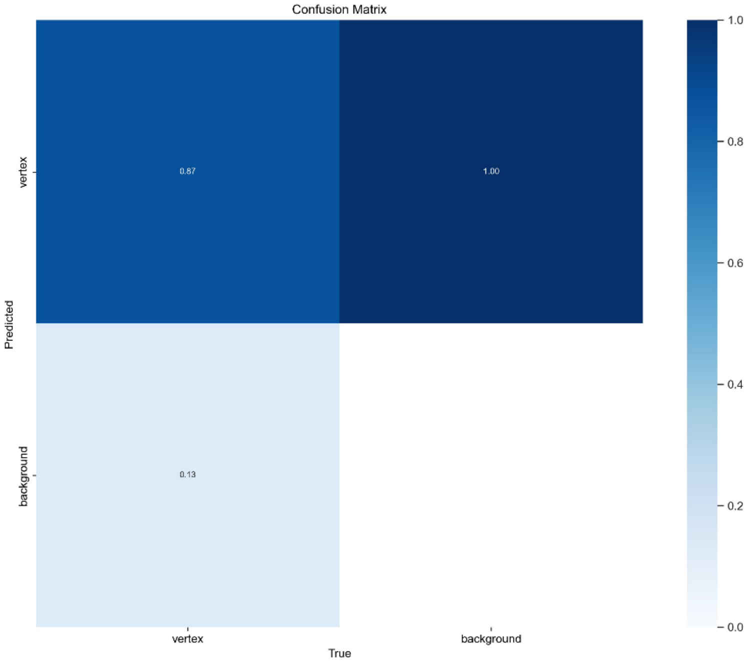

Fig. 8. Error matrix for node segmentation.

In our case, the resulting Precision-Confidence, Precision-Recall and F1 curves look like this and characterize the accuracy of detecting the position of a node in test images in comparison with manually marked positions.

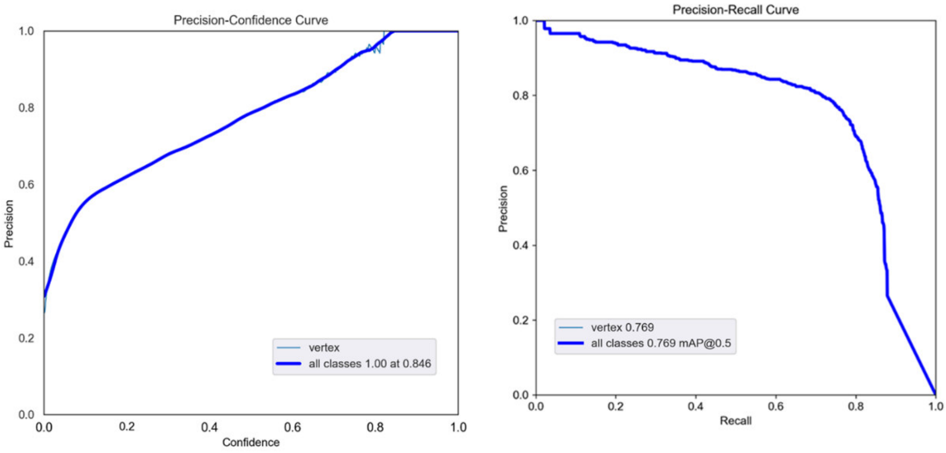

Fig. 9. Precision-Confidence and Precision-Recall curves.

Figure 9 shows that the trained model is characterized by a still high level of false detection of nodes, which is also noticeable in Figures 6-7, when there are false bounding boxes near one of the detected nodes. However, this problem is not significant, since the algorithm for matching vertices and nodes, which works further, combines all closely located vertices into one.

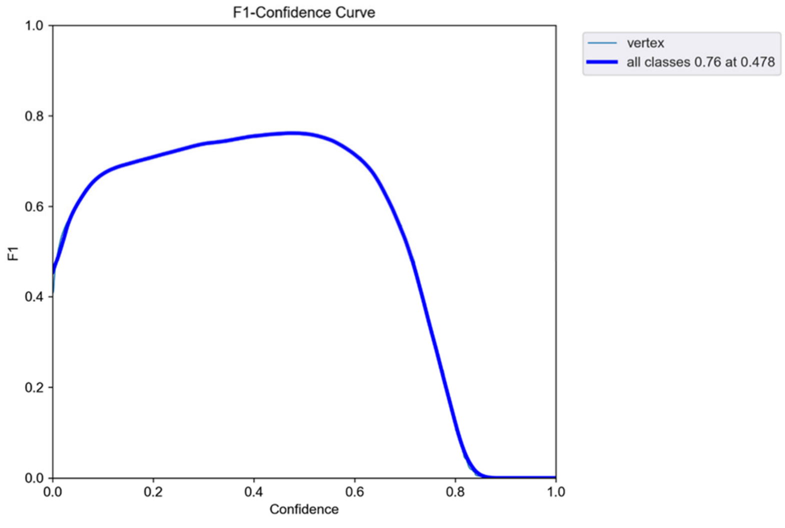

Fig. 10. F1 curve.

F1 Score - Finds the most optimal confidence threshold at which precision and recall produce the highest F1 score. The F1 score calculates the balance between precision and recall. If the F1 score is high, then precision and recall are high, and vice versa. In our case, the curve reaches a maximum value of 0.76 with a confidence value of 0.478.

5 Conclusion

The conducted research allows us to determine methods for segmenting images of cracks and detecting their constituent elements. These methods are based both on the sequential application of threshold binarization, morphological operations, contour search and comparison of nodes with edges based on the minimum distance, and on their combination with node detection using a convolutional neural network. As a result of applying the described methods, it is possible to determine the position of nodes with an accuracy of more than 85% on various images, including blurred and low-contrast ones, and then select edges. These methods can be used to construct a resistance network model based on an undirected weighted graph.

We acknowledge funding from the Russian Science Foundation, Grant No. 23-21-00074.

References

1. R. Gupta, K. D. M. Rao, et al., ACS Applied Materials & Interfaces, **6(16),** 13688-13696 (2014)
2. A.S. Voronin, Y.V. Fadeev, I.V. Govorun, et al., J Mater Sci **56,** 14741-14762 (2021). https://doi.org/10.1007/s10853-021-06206-4
3. A.S. Voronin, et al., Materials **15,** 1449 (2022) https://doi.org/10.3390/ ma15041449
4. D. Li, et al., Journal of Engineering Science and Technology Review **12** 148-156 (2019)
5. Y. Chen, Z. Zhu, Z. Lin, Y. Zhou, Buildings **13,** 1814 (2023)
6. Q. Q. Li, X. L. Liu, Novel approach to pavement image segmentation based on neighboring difference histogram method In: *IEEE 2008 Congress on Image and Signal Processing* (Sanya, China: IEEE, 2008) pp. 792-796.
7. S. J. Schmugge, et al., Crack segmentation by leveraging multiple frames of varying illumination In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (California, USA: IEEE, 2017), pp. 1045-1053.
8. H. N. Nguyen, T. Y. Kam, P. Y. Cheng, Journal of Signal Processing System, **77(3),** 221-240 (2014)
9. L. Zhang, F. Yang, Y. D. Zhang, et al., Road crack detection using deep convolutional neural network In: *2016 IEEE International Conference on Image Processing* (ICIP) (Arizona, USA: IEEE, 2016), pp. 3708-3712
10. J. Valença, D. Dias-Da-Costa, E. Júlio, et al., Measurement, **46(1),** 433-441 (2013)
11. Y. Hu, C. Zhao, Pattern Recognition Research, **1(20103),** 140-147 (2010)
12. K. McCormack, et al., Journal of Computing in Civil Engineering, **30(2),** 1-11 (2016)
13. S. Katsigiannis, et al., J. Build. Eng. **76**, 107105 (2023)
14. L.-C. Chen, et al., Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation In: *Proceedings of the 2018 European Conference on Computer Vision* (ECCV), (Munich, Germany, 8–14 September 2018) pp. 801–818.
15. B. Dwyer, et. al. Roboflow (Version 1.0) [Software]. URL: https://roboflow.com. computer vision.
16. G. Bradski, A. Kaehler, 2008. Learning OpenCV: Computer vision with the OpenCV library, " O'Reilly Media, Inc."
17. G. Jocher, A. Chaurasia, J. Qiu, 2023. YOLO by Ultralytics (Version 8.0.0) [Computer software]. URL: https://github.com/ultralytics/ultralytics
18. L. Shapiro, G. Stockman, *Computer vision* (Prentice Hall, 2001)
19. S. Suzuki et al., Computer Vision, Graphics, and Image Processing, **30(1)** 32-46 (1985)
20. D. M W Powers, Journal of Machine Learning Technologies **2 (1)** 37–63 (2011)