A deep learning approach to crack detection on road surfaces

Roman Sizyakin[a], Viacheslav Voronin[b,c], Nikolay Gapon[c], and Aleksandra Pižurica[a]

[a]Department Telecommunications and Information Processing, TELIN-GAIM, Ghent University, Ghent, Belgium
[b]Moscow State University of Technology "STANKIN", Moscow, Russia
[c]Lab. "Mathematical methods of image processing and intelligent computer vision systems", Don State Technical University, Rostov-on-Don, Russia

ABSTRACT

Currently, modern achievements in the field of deep learning are increasingly being applied in practice. One of the practical uses of deep learning is to detect cracks on the surface of the roadway. The destruction of the roadway is the result of various factors: for example, the use of low-quality material, non-compliance with the standards of laying asphalt, external physical impact, etc. Detection of these damages in automatic mode with high speed and accuracy is an important and complex task. An effective solution to this problem can reduce the time of services that carry out the detection of damage and also increase the safety of road users. The main challenge for automatically detecting such damage, in most cases, is the complex structure of the roadway. To accurately detect this damage, we use U-Net. After that we improve the binary map with localized cracks from the U-Net neural network, using the morphological filtering. This solution allows localizing cracks with higher accuracy in comparison with traditional methods crack detection, as well as modern methods of deep learning. All experiments were performed using the publicly available CRACK500 dataset with examples of cracks and their binary maps.

Keywords: Crack detection on road surfaces, deep learning, machine learning, U-Net, morphological filtering, image segmentation, CRACK500 dataset

1. INTRODUCTION

Over time, the quality of the road surface inevitably deteriorates. Factors that accelerate the degradation of the road surface are: precipitation, significant temperature changes, physical damage, non-compliance with the norms of laying asphalt, etc. These factors lead to the appearance of various damages on the road surface. The most common damages are cracks. The size of road cracks can have a significant variation: from small ones that do not pose a danger to road users, to large ones that can lead to accidents. Detecting cracks, as well as creating maps with their localization, by visual inspection and manual marking by road service employees, is a time-consuming task. If need to create a map of cracks in road surfaces on an urban scale, the process of creating it may take a long time, and as a result, lose its relevance when completed. The solution to this problem can be systems for mapping cracks in automatic mode. Such systems consist of two main components: a video capture device attached to the car, as well as special software. This software includes methods that create a binary map with localized cracks based on the image.

Currently, a significant number of methods have been developed to create a binary map with cracks based on the image. Methods can be roughly divided into three main groups: Methods based on threshold processing, methods based on image filtering, methods based on machine learning. First and earliest group of methods are attributed to the threshold methods of the localization of cracks. The essence of the methods is to select the optimal threshold value that would allow to divide the brightness values of the image in grayscale into two classes: pixels that belong to the background and pixels that belong to foreground objects.[1–4] For this algorithm to work correctly, it is assumed that the image histogram is separable. The disadvantages of this group of methods include the difficulty of selecting a threshold value, as well as numerous false positives. The second group of methods are based on image filtering to detect cracks. These methods are based on the assumption of a priori knowledge of the average size of the crack in the image. In most cases, the methods of this group use morphological filtration, which is confirmed by a number of works.[5–7] The main disadvantage of this group of methods is the need for a
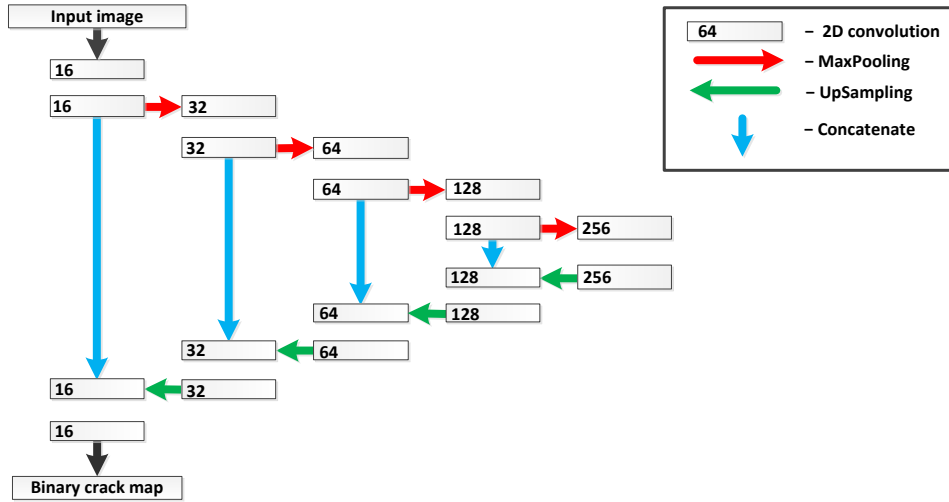
Figure 1: Proposed network architecture

priori knowledge of the size and structure of cracks. The third group of methods is based on the use of machine learning. Traditional machine learning methods include methods that use different texture descriptors to detect road surface cracks.[8–10] Also, several recent works reported using convolution neural network to detect cracks in road surfaces.[11–13] The disadvantages of the third group of methods include the need to use hand-make descriptors, which may limit the applicability of the method in practice. The main disadvantage of methods based on convolutional neural networks is the fixed size of input patches for classification. This restriction in practice leads to two main problems: excessive thickening of the actual crack boundaries on the binary map, as well as inaccurate classification if the size of the classified patch does not match the actual crack size.

In this article, we propose a method based on the latest achievements in image segmentation in the field of deep learning. The proposed method consists of two main stages: the first stage involves creating a preliminary crack map using the U-Net neural network,[14] the second stage is based on morphological filtering of the input image, in order to improve the accuracy of localization of crack boundaries on the preliminary map.

2. PROPOSED METHOD

The general goal of crack detection methods is to create a binary map $d_{i,j}$, called crack map, which accurately describes the location of cracks in the image. The mathematical model of the images containing the cracks of road surface can be represented as follows:

$$Y_{i,j} = (1 - d_{i,j}) \cdot S_{i,j} + d_{i,j} \cdot c_{i,j}, \tag{1}$$

where $Y_{i,j}$ is the image with cracks, $i = \overline{1, I}$ and $j = \overline{1, J}$ are spatial coordinates, with I and J the height and width of the image respectively, $S_{i,j}$ is the crack-free image, $d_{i,j} \in \{0, 1\}$ is the crack map, and where $c_{i,j}$ contains the brightness values of cracks.

To create a preliminary map of cracks we use a convolutional neural network U-Net. The basis of the U-Net neural network is essentially a standard autoencoder, with the only difference that the U-Net network includes intermediate connections between hidden layers. The network architecture used in our work is shown in Figure 1 and has the following layers parameter: C0-3, C1-16, C2-16, C3-32, C4-32, C5-64, C6-64, C7-128, C8-128, C9-256, C10-256, C11-(128+128), C12-128, C13-(64+64), C14-64, C15-(32+32), C16-32, C17-(16+16), C18-16, C19-3, where C denotes a convolutional layer, C0 - denotes a input image, C19- output binary crack map, the first digit corresponds to the layer number and the second to the amount of filters. All convolutional layers have a spatial
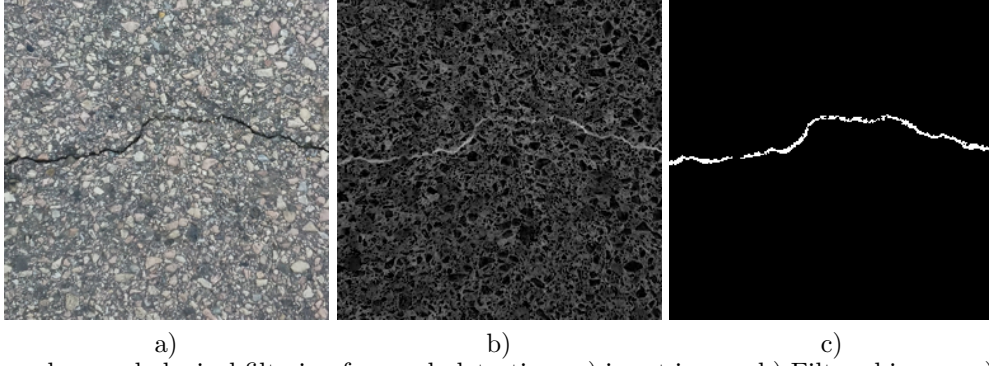
a)   b)   c)

Figure 2: Example morphological filtering for crack detection. a) input image, b) Filtered image, c) Thresholded image

filter size of 3 × 3 pixels. For training in our work we use Adam optimization,[15] with a learning rate of 0.0001. The training process took 75 epochs, with a batch size of 40 pair images/masks.

The U-Net neural network is based on a multidimensional convolution operation, which is defined according to the expression:

$$x_{h,v}^{l,c} = f(\sum_h \sum_v \sum_c x_{h+m,v+n}^{l-1,c} \cdot k_{h,v}^{l,c} + b), \tag{2}$$

where $x_{h,v}^{l,c}$ is the feature map at layer l from modality c, $k_{h,v}^{l,c}$ is the corresponding convolution kernel, $x_{h+m,v+n}^{l-1,c}$ is the feature map from the previous layer, f is the activation function of the hidden layer, and b is a bias.

Using the exponential activation function ELU[16] in comparison with the linear activation function ReLU,[17] allows to achieve convergence of the neural network faster and higher accuracy, as well as exclude the process of batch normalization. The exponential activation function is determined in accordance with the expression:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ a(e^x - 1) & \text{if } x \leq 0, \end{cases} \tag{3}$$

where $a > 0$ is a hyperparameter that controls the value at which the ELU saturates for negative inputs.

For loss estimation we use Sörensen–Dice coefficient,[18,19] showing the measure of the area of correctly marked segments:

$$Loss = \frac{2|x \cap d|}{x + d} \tag{4}$$

where x and d - is estimated and ground truth crack maps, respectively.

2.1 Improved localization of detected cracks

For improved localization of detected road surface cracks after convolutional neural network U-Net, we use multi-scale morphological filtering. Morphological filtering is one of the most common techniques for detecting cracks.[20,21] When detecting cracks, in most cases, two main transformations are used: "Top" and "Bottom" of the hat. These transformations consist of the sequential application of two binary mathematical operations:

$$BottomHat(Y_{i,j}, B) = ((Y_{i,j} \oplus B) \ominus B) - Y_{i,j}, \quad TopHat(Y_{i,j}, B) = Y_{i,j} - ((Y_{i,j} \ominus B) \oplus B), \tag{5}$$

where B is a structuring element, $(Y_{i,j} \oplus B) \ominus B$ is morphological closing, $(Y_{i,j} \ominus B) \oplus B$ is morphological opening, $\ominus$ and $\oplus$ are the erosion and dilation operations, respectively.
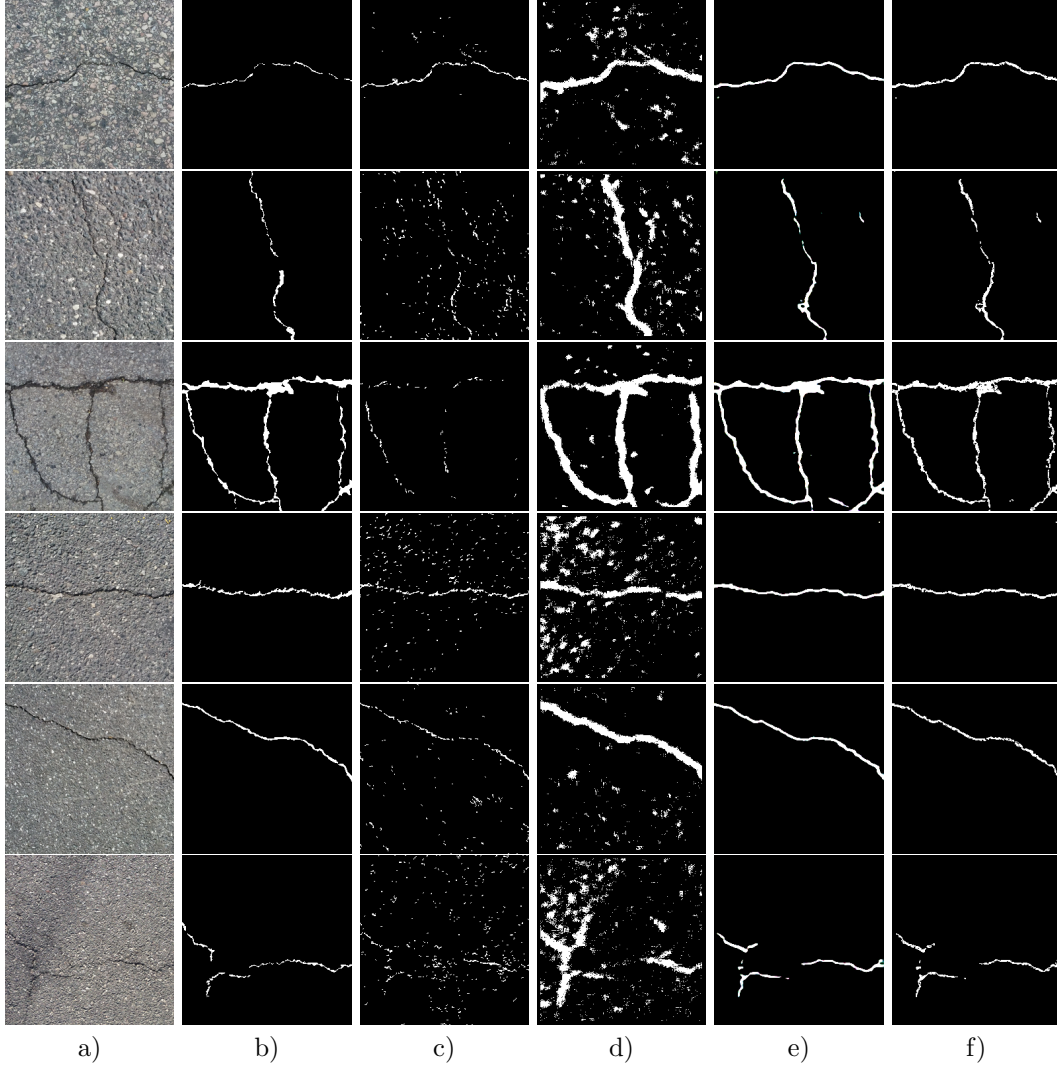
Figure 3: Illustration of the defect detection process (Column from top to bottom): a) source images, b) ground truth crack maps, c) maps of cracks from morphological filtration, d) maps of cracks from DFFN, e) maps of cracks from UNet, f) map of cracks from UNet + enhanced boundaries

The result of morphological filtering is illustrated in Figure 2. The most important parameter responsible for the quality of morphological filtering is the size of the structural element "B". In our case, this parameter is responsible for the" visibility " of the crack in the filtered image. Since the cracks in the road surface have a significant variation in size, we use multiple morphological filtering of the input image using structural elements of different sizes*.[22] After that, the U-Net binary map is multiplied point-by-point by each of the filtered images. At the final stage, the resulting filtered images are converted to binary form using thresholding based on the Otsu method[1] and are combined into a single binary crack map by application of the logical operation "OR".

3. EXPERIMENTAL RESULTS

To test and evaluate the effectiveness of the proposed method, we used a publicly available dataset of marked test images containing cracks in the road surface, CRACK500.[11,23] All images from the dataset were unified and aligned to the same size of 256×256 pixels. The dataset was also divided into two parts: training images (371 image/mask pairs) and validation images (74 image/mask pairs). For comparison the following methods were

---

*A disk-shaped structuring element with a diameter of [1,3,5,7,9] pixels was used

Table 1: Experimental results for the six test images (EB - is an enhanced boundary)

	Method	Recall(R)	False Alarm(FA)	False Miss.(FM)	Precision(P)	F1-meas.(F_1)
Image 1	Morph.Filtering	0.8510	0.0055	0.1490	0.4541	**0.5922**
	DFFN	0.9513	0.0640	0.0487	0.0737	0.1368
	UNET	0.9799	0.0106	0.0201	0.3304	0.4942
	UNET+EB	0.9713	0.0087	0.0287	0.3746	0.5407
Image 2	Morph.Filtering	0.3715	0.0168	0.6285	0.1864	0.2483
	DFFN	0.9316	0.0779	0.0684	0.1104	0.1974
	UNET	0.6568	0.0097	0.3432	0.4127	0.5069
	UNET+EB	0.6196	0.0075	0.3804	0.4613	**0.5289**
Image 3	Morph.Filtering	0.0853	0.0008	0.9147	0.8750	0.1555
	DFFN	0.8694	0.1003	0.1306	0.3666	0.5157
	UNET	0.8991	0.0277	0.1009	0.6845	0.7773
	UNET+EB	0.7843	0.0114	0.2157	0.8218	**0.8026**
Image 4	Morph.Filtering	0.6153	0.0152	0.3847	0.3546	0.4499
	DFFN	0.8630	0.0940	0.1370	0.1106	0.1961
	UNET	0.8368	0.0052	0.1632	0.6838	0.7526
	UNET+EB	0.7820	0.0036	0.2180	0.7446	**0.7628**
Image 5	Morph.Filtering	0.4419	0.0065	0.5581	0.4698	0.4554
	DFFN	0.9301	0.0486	0.0699	0.1999	0.3291
	UNET	0.9455	0.0057	0.0545	0.6826	**0.7928**
	UNET+EB	0.8045	0.0033	0.1955	0.7604	0.7818
Image 6	Morph.Filtering	0.4606	0.0172	0.5394	0.2174	0.2954
	DFFN	0.7964	0.0994	0.2036	0.0767	0.1400
	UNET	0.7370	0.0095	0.2630	0.4448	0.5548
	UNET+EB	0.6880	0.0063	0.3120	0.5322	**0.6001**

used: the method based on morphological filtering, the method based on deep feature fusion,[24] and the standard convolutional neural network U-Net.

Method based on deep feature fusion[24] has 9 hidden convolutional layers and one fully connected layer. The convolutional layers are divided into three main groups: a low-level residual block, a mid-level residual block, and a high-level residual block. Each block include three hidden layers. Each layer in the first block produce 16 feature maps, in the second 32 feature maps, and in the third 64 feature maps. In the hidden layers, a linear rectification unit (ReLU) with batch normalization was applied.[25] The stochastic gradient descent(SGD) method was used for training. The morphological filtering method and the convolutional neural network U-Net correspond to the description given in section 2 of this paper.

To evaluate and compare the performance of methods, we use the following quantitative metrics:

$$FA = \frac{FP}{AlPx - DfPx}, \quad FM = \frac{FN}{AlPx - UdPx} \tag{6}$$

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{7}$$

where FA - probability of false alarm, FM - probability of false missing pixels containing cracks, P - precision, R - recall, F_1 - F_1-measure, TP - true positive, FP - false positive, FN - false negative, $DfPx$ - total amount of pixels belonging to a crack, $UdPx$ - total amount of pixels not belonging to a crack, and $AlPx$ - total amount of pixels in the image.

The results of evaluation of quantitative metrics for 6 random test images from the validation set are collected and shown in table 1.

Based on the analysis of the results obtained (see Figure 3 and Table 1), several conclusions can be drawn. Morphological filtering is an effective auxiliary technique in cases of necessity: pre-localization of cracks,[26,27] creation of virtual modalities,[28] or post-processing, which was demonstrated in this paper. As an independent method for detecting cracks, morphological filtering in most cases will have poor accuracy. Machine learning methods that use input data for classification as a small patch have two important limitations: the first is excessive thickening of the actual crack boundaries, and the second is the inability to see the "whole image", which causes multiple false alarms in areas of the image that are structurally similar to the crack structure. These shortcomings in this work were shown in the DFFN[24] method in this work, and are also found in the works of other authors.[11–13,29] The U-Net[14] convolutional neural network lacks the limitations of patch-based machine learning methods, which is confirmed by the results of table 1. It should also be noted that the combination of the U-Net neural network with post-processing based on morphological filtering allowed us to achieve results superior to the pure U-Net neural network, due to a more precise definition of the boundaries and structure of cracks.

4. CONCLUSION

In this paper, we show the possibility of using the image segmentation method based on the convolutional neural network U-Net in relation to the problem of detecting road surface cracks. To localize cracks with high accuracy, we suggest using multi-scale morphological filtration as post-processing. This solution allowed to increase such quantitative metrics as "Precision" and "F1-measure", with a slight decrease in "Recalls". We also showed the main limitations that can arise when using morphological filtering independently, as well as methods based on patch-based machine learning. Further work will be aimed at modernizing the architecture of the U-Net neural network, which would eliminate the use of post-processing based on morphological filtering.

Acknowledgments

REFERENCES

[1] Otsu, N., "A threshold selection method from gray-level histogram," *IEEE Transaction on Systems, Man, and Cybernatics* **9**, 62–66 (1979).

[2] Kapur, J. N., Sahoo, P. K., and Wong, A. K. C., "A new method for gray-level picture thresholding using entropy of the histogram," *Computer Vision, Graphics, and Image Processing* **29**, 273–285 (1985).

[3] H., O. and L., C. P., "Automatic road crack segmentation using entropy and image dynamic thresholding," *7th European Signal Processing Conference (EUSIPCO)* (2009).

[4] Qingquan, L. and Xianglong, L., "Novel approach to pavement image segmentation based on neighboring difference histogram method," *IEEEE, Congress on Image and Signal Processing* (2008).

[5] Tanaka, N. and Uematsu, K., "A crack detection method in road surface images using morphology," *Proceedings of IAPR Workshop on Machine Vision Applications* , 154–157 (1998).

[6] Gavilan, M., Balcones, D., Marcos, O., Llorca, D. F., Sotelo, M. A., Parra, I., Ocana, M., Aliseda, P., Yarza, P., and Amirola, A., "Adaptive road crack detection system by pavement classification," *MDPI, Sensors* , 9628–9657 (2011).

[7] Oliveira, H. and Correia, P. L., "Road surface crack detection: improved segmentation with pixel-based refinement," *25th European Signal Processing Conference (EUSIPCO)* , 2080–2084 (2017).

[8] Oliveira, H. and Correia, P., "Supervised strategies for cracks detection in images of road pavement flexible surfaces," *IEEE, 16th European Signal Processing Conference* , 1–5 (2008).

[9] Lin, J. and Liu, Y., "Detection based on svm in the pavement distress image," *IEEE, Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science* , 544–547 (2010).

[10] Hu, Y., Zhao, C.-X., and Wang, H.-N., "Automatic pavement crack detection using texture and shape descriptors," *IETE Technical Review* , 398–405 (2010).

[11] Lei, Z., Fan, Y., Yimin, D., and Ying, J. Z., "Road crack detection using deep convolutional neural network," *IEEE International Conference on Image Processing (ICIP)* , 3708–3712 (2016).

[12] Cha, Y.-J., Choi, W., and Büyüköztürk, O., "Deep learning-based crack damage detection using convolutional neural networks," *Computer - Aided Civil and Infrastructure Engineering* **32**, 361–378 (2017).

[13] Kim, B. and Cho, S., "Automated vision-based detection of cracks on concrete surfaces using a deep learning technique," *MDPI and ACS Style* (2018).

[14] Ronneberger, O., Fischer, P., and Brox, T., "U-net: convolutional networks for biomedical image segmentation," *Springer,Medical Image Computing and Computer-Assisted Intervention, MICCAI* **9351** (2015).

[15] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *ICLR: International Conference on Learning Representations* (2015).

[16] Clevert, D., Unterthiner, T., and Hochreiter, S., "Fast and accurate deep network learning by exponential linear units (ELUs)," *ICLR: International Conference on Learning Representations* (2016).

[17] Glorot, X., Bordes, A., and Bengio, Y., "Deep sparse rectifier neural networks," *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR* **15**, 315–323 (2011).

[18] Sörensen T., A., "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Kongelige Danske Videnskabernes Selskab* **5**(4), 1–34 (1948).

[19] Dice Lee, R., "Measures of the amount of ecologic association between species," *Ecology* **26**(3), 297–302 (1945).

[20] Sizyakin, R., Cornelis, B., Meeus, L., Martens, M., Voronin, V., and Pižurica, A., "A deep learning approach to crack detection in panel paintings," *Image Processing for Art Investigation (IP4AI)* , 40–42 (2018).

[21] Sizyakin, R., Cornelis, B., Meeus, L., Dubois, H., Martens, M., Voronin, V., and Pižurica, A., "Crack detection in paintings using convolutional neural networks," *IEEE Access* **8**, 74535–74552 (2020).

[22] Cornelis, B., Ružić, T., Gezels, E., Dooms, A., Pižurica, A., Platiša, L., Cornelis, J., Martens, M., Mey, M. D., and Daubechies, I., "Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent altarpiece," *Signal Processing* **93**(3), 605–619 (2013).

[23] Yang, F., Zhang, L., S.Yu, Prokhorov, D., Mei, X., and Ling, H., "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE, Transactions on Intelligent Transportation Systems* , 1–11 (2019).

[24] Song, W., Li, S., Fang, L., and Lu, T., "Hyperspectral image classification with deep feature fusion network," *IEEE Transactions on Geoscience and Remote Sensing* **56**, 3173–3184 (2018).

[25] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on International Conference on Machine Learning* , 448–456 (2015).

[26] Giakoumis, I., Nikolaidis, N., and Pitas, I., "Digital image processing techniques for the detection and removal of cracks in digitized paintings," *IEEE Transactions on Image Processing* **15**, 178–188 (2006).

[27] Spagnolo, G. S. and Somma, F., "Virtual restoration of cracks in digitized image of paintings," *International Conference on Defects in Insulating Materials, Journal of Physics* **249** (2010).

[28] Sizyakin, R., Voronin, V., Gapon, N., Zelensky, A., and Pižurica, A., "Automatic detection of welding defects using the convolutional neural network," *SPIE, Automated Visual Inspection and Machine Vision III* (2019).

[29] Li, Y., Li, H., and Wang, H., "Pixel-wise crack detection using deep local pattern predictor for robot application," *MDPI and ACS Style* (2018).