Machine learning based detection and deep learning based image inpainting of preparation artefacts in micrographs

A. Jansche, A. K. Choudhary, T. Bernthaler and G. Schneider

Materials Research Institute, Aalen University, Beethovenstr. 1, D-73430 Aalen, Germany,
andreas.jansche@hs-aalen.de

ABSTRACT

Artefacts like stains or scratches introduced during the metallographic sample preparation process are a problem for manual as well as automatic image analysis. We show the use of image inpainting – normally used for reconstruction of missing parts of an image – for removal of these artefacts from micrographs. Light optical microscopy images of non-metallic inclusions in steels are used to demonstrate the results. We build on top of earlier work, in which the segmentation and classification of oxides, sulphides and artefacts was demonstrated. The masks produced from the segmentation model were re-used in the image inpainting step to mask out only regions which show artefacts. While the approach produced good results in terms of qualitative inspection / human perception the quantitative results suggested that the analysis was biased towards missing some inclusions as they were removed by the inpainting. We conclude by discussing steps to further improve the presented results.

Keywords: image inpainting, convolutional neural networks, partial convolutions, quantitative microstructure analysis, non-metallic inclusions

1 INTRODUCTION

Automated quantitative microstructure analysis (QMA) as well as qualitative inspection of samples by experts are essential tools for materials science as well as industrial applications of materials microscopy like quality assurance. In order to get reliable results based on the microstructure of materials the metallographic preparation of the samples has to be of high standard or at least good enough to not interfere with the parts of the microstructure one intends to analyze. However, achieving preparation free of artefacts over large areas is often not possible due to the challenging nature of the materials or time constraints in industrial settings. This work extends a workflow based on previous work [1] for the analysis of non-metallic inclusions that includes the segmentation of micrographs by pixel-wise classification using a multi-layer perceptron and RGB features and object classification to distinguish oxides, sulphides and preparation artefacts using a random forest classifier and Haralick texture features. In this work we use convolutional neural networks and partial convolutions for image inpainting [2] to remove preparation artefacts as well as other segmented objects from the micrograph. The first two steps of the workflow provide a binary image for each class which can be used to mask the relevant areas for the inpainting step. The inpainted images can then be presented to experts without distracting artefacts or only showing certain objects/phases. It is also investigated if the inpainted images can be used as artefact-free images for QMA methods that analyze non-metallic inclusions and how it affects the results. While the latter potentially enables more reliable QMA results the former supports the viewer during a qualitative assessment of the sample as only the relevant parts of the micrograph are shown. This allows an unobstructed view on certain objects or phases and their size, morphological features or distribution within the sample. Even though the approach is presented for non-metallic inclusions we believe it is transferable to other materials and microstructures.

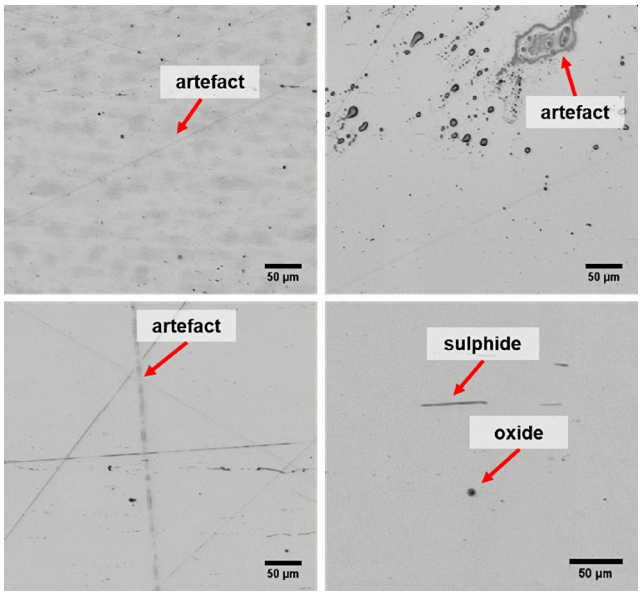

Figure 1: While analyzing steel samples for non-metallic inclusions preparation artefacts will prevent the QMA of oxides and sulphides using established methods as well as obscure the view of experts.

TechConnect Briefs 2021, TechConnect.org, ISBN 978-0-578-99550-2

2 RELATED WORK

The task of image inpainting is still a very challenging computer vision task. There is a wide range of available algorithms for this problem, including traditional image processing methods as well as machine learning based methods like convolutional neural networks, including generative adversarial networks. Even though significant improvements were made with the deep learning methods (often based on very complex architectures), results still suffered from unsharp regions, shift in color, unrealistic reconstructions or other problems. [3]

In 2018, Liu et al. introduced partial convolutions and their application for image inpainting in [2]. This method outperformed other methods[1], which didn´t produce realistic results and/or relied on computationally expensive postprocessing to compensate for image artefacts.

Partial convolutions were then outperformed by other methods. In [4] Yu et al. generalized the approach and extended it with gated convolutions and a new loss function.

Other approaches tend to include additional features alongside the image. For example, Nazeri et al. proposed a two-step approach where a first network tries to complete the edge map of the input image and a second network then uses the edges to achieve improved inpainting results [5]. Similarly, Wu et al. trained a network to predict the local binary patterns (LBP) of the missing image regions. These predicted texture features are then used as an additional input for a second network to do the inpainting step [6].

3 PARTIAL CONVOLUTIONS

The core component of the used image inpainting approach are partial convolutional layers, introduced by Liu et al. in [1]. As shown in (1), only unmasked parts of the input image *X* influence the result of the convolutional operation, as the other parts are canceled out by the element-wise multiplication with the binary mask *M*. The different amount of masked pixels is accounted for by applying *sum(1)/sum(M)* as a scaling factor. (*1* denotes a matrix with the same dimensions of *M*, but all elements having the value of 1.)

$$x' = \begin{cases} \mathbf{W}^T(\mathbf{X} \odot \mathbf{M})\frac{1}{\mathrm{sum}(\mathbf{M})} + b, & \text{if } \mathrm{sum}(\mathbf{M}) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In addition to the partial convolution operation itself, an update of the mask is carried out after each partial convolution. As shown in (2), the mask gets updated with a value of 1 if at least one unmasked pixel was taken into account in the partical convolution before (indicated by *sum(M)* being greater than 0).

[1] *The method was considered state of the art when this work was carried out and submitted in 2019.*

$$m' = \begin{cases} 1, & \text{if } \mathrm{sum}(\mathbf{M}) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

This way the mask gets smaller after every operation in which a masked part of the image was processed.

4 EXPERIMENTAL SETUP

The dataset consists of 146 image patches with a size of 512x512 pixels each. 106 of these images contain at least one preparation artefact while 40 images show no artefacts.

For each of the images the dataset contains a corresponding binary mask that marks the area of the image that contain artefacts. These masks were generated using the segmentation model from [1].

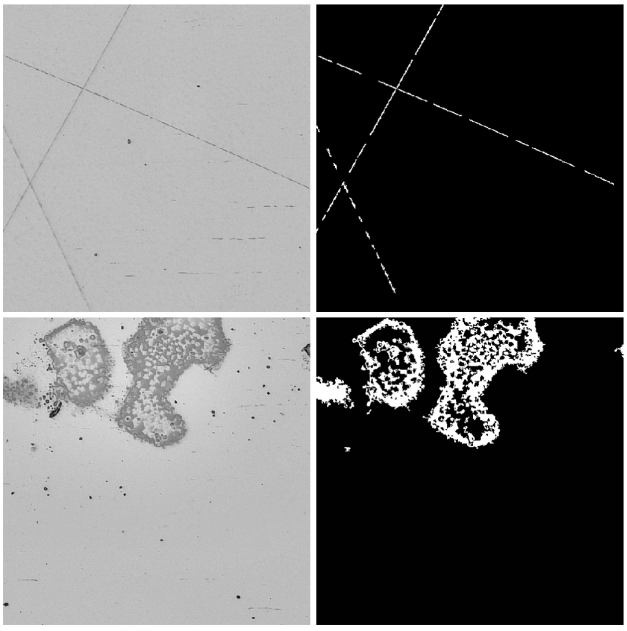

Figure 2: Example images from the dataset. On the left two image patches are shown. Both show artefacts in the form of scratches (top) or chemical stains (bottom). On the right the binary masks for each image are shown.

For model training, only the images without artefacts were used. They were split into 32 training images, 4 validation images and 4 test images. Data augmentation was applied during model training to effectively increase dataset size.

The architecture of the model used is derived from the structure of a U-Net. The convolutional layers are replaced with partial convolutional layers. Masks are passed through the network alongside the input image. The skip connections of the U-Net allow unmasked pixels to be copied to the decoder part of the network as well as to the final output image.

Initial weights are used from a VGG16, pre-trained on ImageNet. Our implementation was based on [7].

Peak signal-to-noise ratio (PSNR) is used as performance metric. The loss function is a combination of different per-pixel, perceptual, style and variation loss functions. Details can be found in [2] and [7]. The model was trained with a batch size of 16 for 300 epochs with a total of 30,000 steps. Training was carried out in four parts, two with and two without batch normalization, similar as referred to in [7].

Because only images free of artefacts were used, the model was forced to learn to inpaint only ideal microstructures as opposed to producing artefacts itself. Random masks (as shown in figure 3 and implemented in [7]) were created on-the-fly during training and applied on the images.

Figure 3: Three examples of random masks used for training with artefact-free images.

Before applying the model to the images with artefacts, the masks obtained from the segmentation model from [1] are dilated using a 9x9 pixel structuring element. This is done to compensate for the often either very small or sometimes incomplete segmentation results.

5 RESULTS

Qualitative results of the inpainting are judged by visual inspection. For small or medium sized artefacts, the inpainting works seamlessly and without introducing noticeable distortions in the image. Only for larger inpainted areas (compare the examples c) and e) in figure 4) a mismatch in color / brightness and a weak checkered pattern can be observed.

Quantitative results were obtained using our segmentation model from [1] to calculate the phase fraction of oxides, sulphides and steel for each image in the dataset. It is important to note that the results for the original images (figures 6 and 7) show many more outliers in the direction of higher phase fractions. Those were only removed for better visibility in the figures. This indicates that even though the object classification was carried out in [1], some objects were falsely left in the class of oxides or sulphides (partly as they were overlapping with artefacts).

Also, for both, oxides and sulphides, quantitative results show lower (average) phase fractions overall.

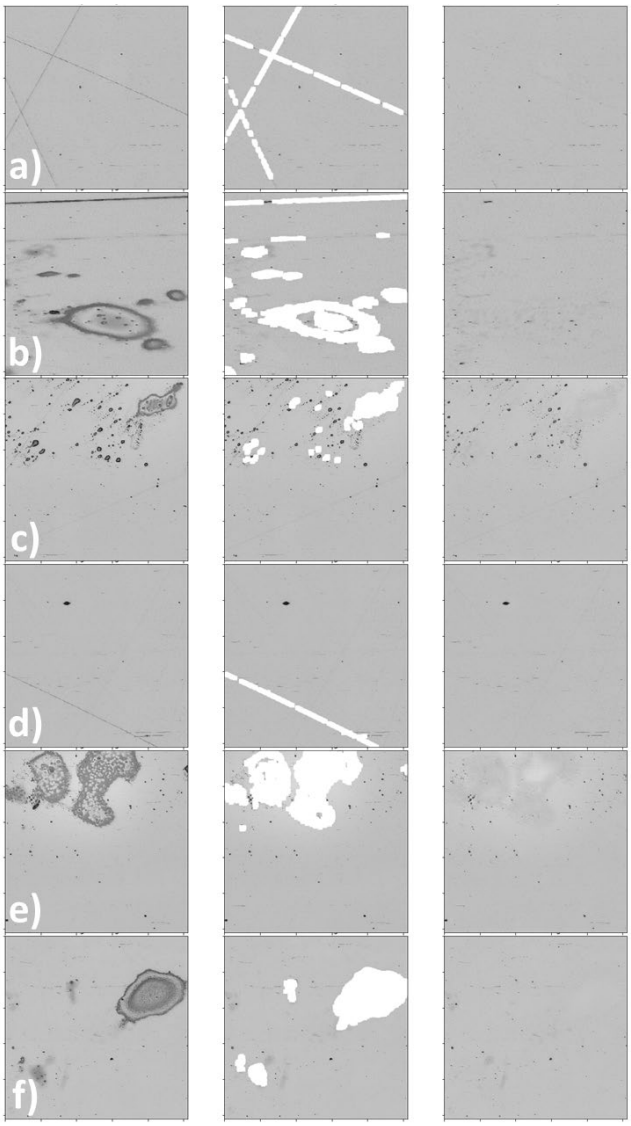

Figure 4: Visualization of results. From left to right, the figure shows the original images (including artefacts), the images for which the segmentation results obtained from [1] where used to mask the artefacts and the final inpainted result images with removed artefacts.

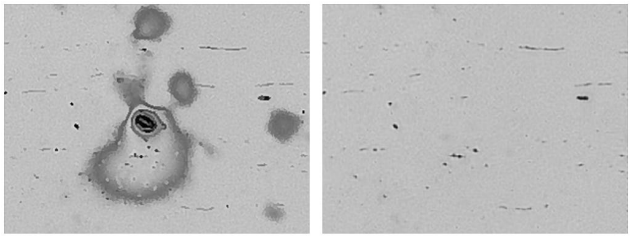

Figure 5: Example of an outlier showing a big difference in quantitative results. The masks are so big, that they also cover actual inclusions which are then missing in the quantitative analysis.

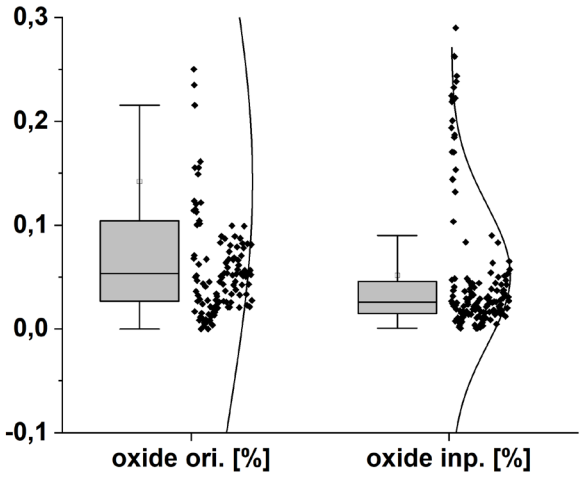

Figure 6: Phase fractions for oxide in the original images (left) and the inpainted images (right).

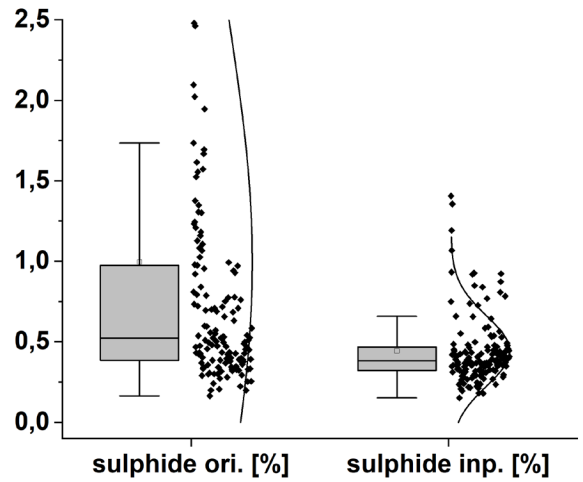

Figure 7: Phase fractions for the sulphide in the original images (left) and the inpainted images (right).

6 DISCUSSION

Qualitative results were very good for the vast majority of the images. While slight distortions of the image are present if big areas of the image are inpainted, they don´t hinder the manual inspections of the non-metallic inclusions. The results could be improved further with newer inpainting approaches like [4,5,6], which yield very good results on other datasets.

The tendency of the current approach to remove actual sulfides and oxides along with the preparation artefacts limits the usability for quantitative analysis. Especially for extreme cases of artefacts in terms of size or frequency the method has to be improved and results have to be analyzed further. Implementing an optimized preprocessing of the masks to only dilate the smallest masks and thereby reducing the area which is inpainted may already improve results. To make a final conclusion about the usability for QMA on an object level (as in quantifying the size and morphology of each individual inclusion) further work would need to be carried out. We would expect that the only disadvantage would be that fewer inclusions would be analyzed, which would even out if done for larger sample areas.

The application of this approach to other, more complex microstructures has also been identified as potential future work.

7 ACKNOWLEDGEMENTS

This work was funded by the Federal Ministry of Education and Research of Germany within the KLEVER project (13FH255PA6) / FHprofUnt program. We also want to thank Florian Trier for his support in preparation of the manuscript.

REFERENCES

[1] Choudhary, Amit Kumar; Jansche, Andreas; Badmos, Olatomiwa; Bernthaler, Timo and Schneider, Gerhard (2019): A machine learning classification model to identify non-metallic inclusions and preparation artefacts in steel alloys. In: Fortschritte in der Metallographie. Vortragstexte der 53. Metallographie-Tagung, 18.-20. September 2019 in Dresden, Bd. 53 (Sonderbände der Praktischen Metallographie), S. 215–220.

[2] Liu, Guilin; Reda, Fitsum A.; Shih, Kevin J.; Wang, Ting-Chun; Tao, Andrew and Catanzaro, Bryan (2018): Image Inpainting for Irregular Holes Using Partial Convolutions. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, Yair Weiss (Eds.): Computer Vision – ECCV 2018, vol. 11215. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 89–105.

[3] Jam, Jireh; Kendrick, Connah; Walker, Kevin; Drouard, Vincent; Hsu, Jison Gee-Sern; Yap and Moi Hoon (2021): A comprehensive review of past and present image inpainting methods. In Computer Vision and Image Understanding 203, p. 103147. DOI: 10.1016/j.cviu.2020.103147.

[4] Yu, Jiahui; Lin, Zhe; Yang, Jimei; Shen, Xiaohui; Lu, Xin and Huang, Thomas S. (2018): Free-Form Image Inpainting with Gated Convolution. In CoRR abs/1806.03589.

[5] Nazeri, Kamyar; Ng, Eric; Joseph, Tony; Qureshi, Faisal Z. and Ebrahimi, Mehran (2019): EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning. In CoRR abs/1901.00212.

[6] Wu, Haiwei; Zhou, Jiantao; Li, Yuanman (2020): Deep Generative Model for Image Inpainting with Local Binary Pattern Learning and Spatial Attention. In CoRR abs/2009.01031.

[7] Gruber, Mathias (2019): Partial Convolutions for Image Inpainting using Keras. Unofficial implementation of "Image Inpainting for Irregular Holes Using Partial Convolutions". github.