Universal Semantic-less Texture Boundary Detection for Microscopy (and Metallography)

Matan Rusanovsky
NRCN
Israel
matanr@nrcn.org.il

Ofer Be'eri
NRCN
Israel
ofer.beeri@gmail.com

Shai Avidan
Tel-Aviv University
Israel
avidan@tauex.tau.ac.il

Gal Oren
Technion, NRCN
Israel
galoren@cs.technion.ac.il

Abstract

The automated analysis of textures has always been a topic of importance in metallographic imaging in particular and in microscopy in general. Those analyzed textures are used in a variety of applications, and as such, texture analysis is at the backbone of most, if not all, other vision tasks. However, the task of texture analysis greatly differs from related and well-defined tasks such as edge, contour, and semantic analysis for detection and segmentation. As texture perception is hard even for humans to define and includes a subjective outlook, computerized texture-based segmentation in semantic-less and texture-oriented images has not been achieved so far. Moreover, it is difficult to apply recent computer vision algorithms to such images because of database shortages in this domain, as well as a shortage of accurately labeled data.

Therefore, we wish to develop a *Universal Texture Representation* (UTR). This representation will allow us to segment *any* texture image in *any* domain and even develop a *Universal Texture Boundary Detector* (UTBD). Crucially, such algorithms should work on texture images (and even videos) that have no semantic meaning, such as metallographic textures; hence, the vast literature on edge, contour, and semantic segmentation can not be used as is in our context.

Henceforth, we formulate and define our problem: Universal semantic-less texture boundary detection. A solution to this newly defined problem – which, in this work, we present the initial path towards on our Texture Boundary in Metallography (TBM) dataset – could be used in a variety of applications as is or as an enhancer to other closely related vision tasks. For example, it could help quickly segment new images based on single-click segmentation cues provided by the user, or it could help retrieve images with similar textures from past experiments.

1 Introduction – The Significance of Universal Texture Representation

The automated analysis of textures has always been a topic of importance in biomedical [10] and metallographic [5] imaging, involving different acquisition techniques, from simple microscopes to scanning electron microscope (SEM), and from elementary X-ray radiography and ultrasound (US), to computed tomography (CT), positron emission tomography (PET) and magnetic resonance imaging (MRI). Those analyzed textures are used in a variety of applications, including the segmentation of different anatomical areas [14] or crystalline orientation [41], the differentiation between normal and pathological subjects/inclusions/precipitates, as well as the classification and grading of a large number of pathological conditions or mechanical [46, 38, 4] and electrical [50, 22, 2] properties. As such, texture analysis is at the backbone of most, if not all, other vision tasks [49]. However, it also

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

greatly differs from related and well-defined ones such as edge, contour, and semantic analysis for detection and segmentation [36]. As texture perception is hard even for humans to define and includes a subjective outlook, computerized texture-based segmentation in semantic-less and texture-oriented images (such as microscopy) has not been achieved so far. Moreover, it is difficult to apply recent computer vision algorithms to such images because of database shortages in this domain, as well as a shortage of accurately labeled data.

Therefore, we wish to develop a *Universal Texture Representation* (UTR) that will allow us to segment *any* texture image in *any* domain. Going one step further, we also wish to develop a *Universal Texture Boundary Detector* (UTBD). Such a detector will be able to distinguish between different textures without having to consistently assign the same label to the same texture in different parts of the image. For example, it could help quickly segment new images based on single-click segmentation cues provided by the user. Or, it could help retrieve images with similar textures from past experiments for comparison, such as in [12, 13].

Crucially, our algorithms should work on texture images that have no semantic meaning; hence, the vast literature on edge/contour/semantic segmentation can not be used as is in our context, including Segment Anything Models (SAM) [29], that might succeed in cases in which semantics appears in the image, but fails when those are missing or form a texture collaboratively. Moreover, as the problem of boundary detection in images of new and different domains (medical, remote sensing, material inspection, etc.) is crucial, we would like to develop self-supervised or few-shot learning methods that can learn to detect boundaries quickly on new domains.

We stress that in our context, there is a clear distinction between segmentation and boundary detection. In segmentation, the algorithm must be consistent in that the same texture in different image regions must be consistently labeled with the same label. In contrast, in boundary detection, all that is needed is to detect the boundary between two different textures. As such, there is a need for new assessment methods that will allow distinction between closely related (and occasionally overlapping) vision tasks, with special emphasis on quantitative measures for semantic-less boundary detection, such as the performances of the new tools given differential perspectives and textural-oriented image-sets.

Henceforth, we formulate and define our problem: Universal semantic-less texture boundary detection. A suitable solution to the problem (section 4) must successfully detect boundaries of unseen textures without regard to semantic meaning. The solution will be required to learn and generalize to detect texture boundaries on images of videos from single or multiple datasets and to be tested across domains on out-of-distribution textures. The test benchmark dataset will be formed from known texture datasets, including ones that are specific for microscopy and metallography (section 3).

2 Background – Texture Synthesis, Analysis, and Boundary Detection

Although texture analysis is an important research area in computer vision, there is no precise definition of the notion of texture. The main reason is that natural textures often display different yet contradicting properties, such as regularity versus randomness and uniformity versus distortion, which can hardly be described in a unified manner.

Following these properties, textures can be schematically divided into two main categories by their degree of randomness. Generally, the complexity of texture analysis corresponds to the degree of randomness imposed within it. A regular texture is formed by regular tiling of easily identifiable small primitives organized into strong periodic patterns. On the other hand, a stochastic texture exhibits less noticeable elements and displays rather random patterns.

The research on texture perception dates back to the seminal work of Julesz [26], who conjectured that humans cannot distinguish between textures with identical second-order statistics. He proved his conjecture wrong [25], though the concept that image textures could be modeled based on low-order statistics remained. He later introduced *textons* as the basic units of texture [7].

Texture synthesis, extensively applied in image editing and computer games, employs classic frequency domain analysis and synthesis techniques such as those introduced by Heeger and Bergen [20], De Bonet [9], and Portilla and Simoncelli [39], which utilize statistical properties to generate new textures, while non-parametric approaches, including methods by Efros and Freeman [16], Kwatra et al. [31], and others [48, 44, 6], achieve texture synthesis by sampling patches from the input texture image and matching their distribution to the input texture.

Deep learning for texture synthesis by Gatys *et al.* [18] follows the approach of Heeger and Bergen [20]. Instead of matching the histograms in the image pyramid, they match the Gram matrix of different feature maps of the texture image, where the Gram matrix measures the correlation between features at selected layers of a neural network. Subsequent works later improved this approach [45, 42].

Alternatively, one can use a generative adversarial network (GAN) to synthesize textures that resemble the input exemplar. Li and Wand [32] used a GAN combined with a Markov Random Field to synthesize texture images from neural patches. Liu [35] improved the method of Gatys [18] by adding constraints on the Fourier spectrum of the synthesized image, and Li [34] use a feed-forward network to synthesize diversified texture images. Zhou [51] use GANs to spatially expand texture exemplars, extending non-stationary structures. Frühstück [17] synthesize large textures using a pre-trained generator, which can produce images at higher resolutions.

Texture analysis is often associated with semantic segmentation and can be roughly divided into two parts: Semantic Segmentation and Instance Segmentation. In Semantic Segmentation, each pixel is assigned a label of the class (i.e., road, vehicle, sky, person, etc.) it belongs to. In Instance Segmentation, each pixel is assigned a label of the class and the instance it belongs to (i.e., two different people will be assigned different labels).

Recently, the two have been combined into a single, holistic approach, termed Panoptic Segmentation, where the goal is to assign each pixel its class label, and in case there are multiple instances of the object, assign the different objects a different label. Panoptic Segmentation is used for various applications such as Autonomous Driving and general image understanding.

Another closely related branch of texture analysis is that of **Boundary Detection**, where the goal is to detect meaningful boundaries/interfaces between textures, cells, crystallites, etc. This branch evolved from the problem of edge detection, where edge detection is concerned with detecting changes in intensity values, while boundary detection is concerned with detecting boundaries between regions (i.e., textures). Traditional methods mostly focus on low-level cues. For example, Martin *et al.* [37] proposed a probabilistic boundary (Pb) detection module that learns to detect boundaries in images using local brightness, color, and texture cues. Dollár and Zitnick [15] use Structured Forests to map local image patch to local edge maps. A long line of work focused on extending the local approach to take global cues into consideration as well.

A sequence of papers [8, 43, 30] suggested using deep features to improve image boundary detection. Common to all is the use of deep features as the space in which to detect boundaries. Exploiting multi-scale representations and dilated convolutions helped He *et al.* [19] push the state-of-the-art in the field even further. Recently, Pu *et al.* adapted Transformers for boundary detection [40]. Their approach is based on a two-level Transformer architecture, where the first level captures global scene information, and the second refinement level calculates the boundaries. Their network achieves state-of-the-art results on several datasets [24], including the Berkley Segmentation Dataset [3]. The latest advancement using Transformers for this task is the Segment Anything Model (SAM) [29], which presents a significant step forward while also having limitations in professional data, especially in textural and medical imaging [23, 21].

Common to all these methods is the strong connection they make between object boundaries and boundary detection. We, on the other hand, are interested in Markov Boundary Detection. This means we do not care about global (non-Markov) scene information when making a local boundary detection decision. This also means we are not interested in segmentation because segmentation implies consistent labeling. That is, a segmentation algorithm should assign the same label to the same texture anywhere in the image plane. We, on the other hand, are only interested in the existence of boundaries between textures.

3 Dataset – Texture Boundary in Metallography (TBM)

Metallography analysis offers a wonderful test bed to develop novel Computer Vision algorithms for texture analysis and synthesis. We believe such algorithms can be transferred as is or with slight modifications to other domains in Computer Vision and beyond. Thus, aside from the common work on current relevant general image segmentation datasets (such as MMSegmentation, BSDS500, ADE20K, CityScape, Camouflage, Brodatz, Real World), we construct a database, namely Texture

Boundary in Metallography (TBM)[1], that is as general as possible, with annotations that correspond to the task of texture boundary detection, in microscopy, and especially in metallography. This database serves as a benchmark for our research and the following future works to establish the direct contribution to the needed sciences. In TBM, metallographic images of approximately 1.2 mm × 0.8mm were used, while multiple experts in material science tagged each image to verify for no annotator bias. Specifically, in TBM, we have created a dataset for metallographic texture boundary detection, consisting of cropped squares (128 × 128 pixels) of said metallographic scans with corresponding expert manual tags of grains' boundary as ground truth (320 images).

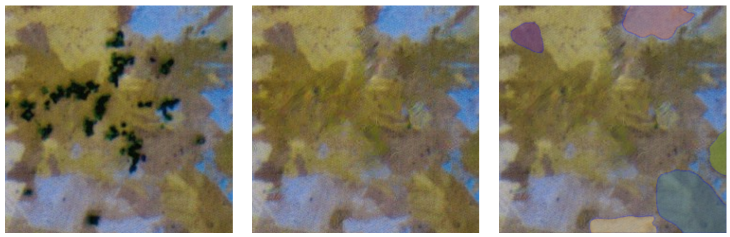
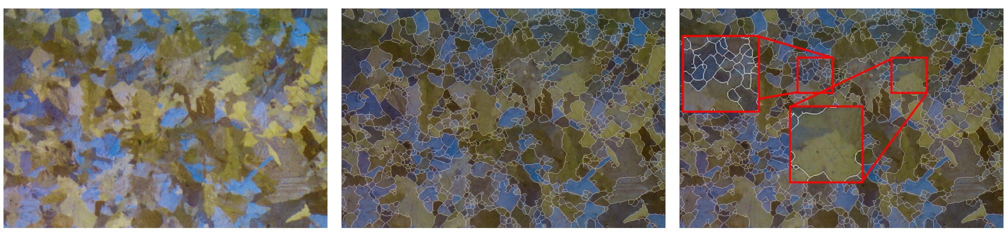


Figure 1: Metallographic textures (from left): Zoomed metallography with impurities; Removed impurities using deep inpainting; and failed semantic segmentation with Segment Anything [29].



(a) Metallographic input image. (b) Supervised GB SS. (c) Wrong predictions (magnified).

Figure 2: Metallographic image input and the supervised GB semantic segmentation.

4 Preliminary Results and Path to Solution

Preliminary Results: An end-to-end computer vision approach for quantitative metallography was presented in [41]. [41] presented an integrated solution for quantifying the quality of a metallographic image using deep semantic segmentation for both impurities and grain boundary, generative impurities inpainting, and impurities anomaly detection. However, the algorithms in that work assume an extensive annotated database that was supervised by material scientists and was tailored specifically for that work, such as TBM. It was used for semantic segmentation using U-Net, with limited accuracy results. Our suggested research complements this work, as it relaxes the task-specific data assumption and considers a limited to zero supervision UTBD approach for different microscopy inference tasks.

Consider, for example, Figure 1. It shows a microscopy image of a complicated alloy from [41] after the removal of the impurities using deep inpainting and a failed attempt to use alleged universal segmentation models [29]. We aim to detect the boundaries between the different textures in the image (and in other images of the same alloy with other different textures. For example, Figure 2a). Figure 2b shows the result of texture boundary detection on the image from Figure 2a using the previously mentioned state-of-the-art semantic segmentation tool, tailored for metallography [41]. A sensitive eye will notice that while many edges were detected correctly, others were not (Figure 2c). Of course, this is due to the fact that such images are texture but not edge or semantically oriented.

To evaluate the context role in modern boundary detection, we ran EDETR [40], an edge detection transformer model [24] in two settings. In both cases, we randomly select windows of size 100×100. In the first one, we ran EDETR on the entire image but evaluated the algorithm's performance only on the random window. In the second setting, we placed the randomly cropped window in the center

[1]Texture Boundary in Metallography (TBM) dataset: https://doi.org/10.5281/zenodo.8386997

of an otherwise completely black image and then ran EDETR. In both cases, we used EDETR, which was trained on the training set of BSDS500 and tested on the images of the test set.

During the evaluation, three key metrics are employed to assess performance: fixed contour threshold (ODS), per-image best threshold (OIS), and average precision (AP). Results are shown in Table 1a. As can be seen, context plays an important role in the algorithm's performance. A few examples can be seen in Figure 3. The same experiment was repeated on the TBM dataset of 128×128 images, with 60×60 windows. The results are shown in Table 1b, and examples are shown in Figure 4.

Setting	ODS	OIS	AP
EDETR $\rightarrow$ Crop	0.684	0.764	0.680
Crop $\rightarrow$ EDETR	0.589	0.274	0.582

(a) BSDS500 dataset natural images.

Setting	ODS	OIS	AP
EDETR $\rightarrow$ Crop	0.813	0.831	0.189
Crop $\rightarrow$ EDETR	0.393	0.412	0.310

(b) TBM dataset textural microscopy images.

Table 1: Two experiments on the rule of context in texture boundary detection.

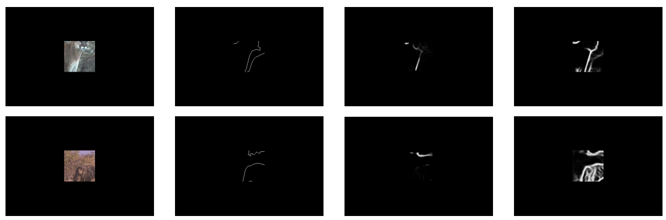


Figure 3: Examples of context importance in EDTER that was trained on BSDS500. From left to right: input window, ground truth, crop followed by EDTER, EDTER on full image followed by crop.

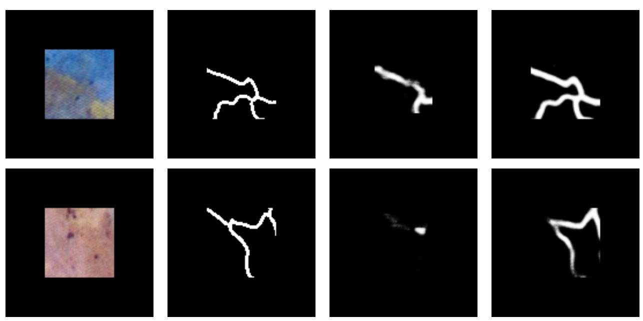


Figure 4: Examples of context importance in EDTER that was trained on metallographic database. From left to right: input window, ground truth, crop followed by EDTER, EDTER on full image followed by crop.

Path to Solution: Our proposed approach will center on developing a UTR through the utilization of a pre-trained Vision Transformers (ViT) inpainter model [33, 47]. The UTR will aim to impart the model with a profound understanding of textures, completely independent of semantic constraints. The pre-training phase will leverage the DTD dataset [11], an extensive texture database encompassing 5640 images spanning 47 diverse texture categories. During this phase, the inpainter model will learn to comprehend an array of textures in a semantic-less fashion, devoid of specific object or label associations. Subsequently, fine-tuning will be carried out to adapt the model for UTBD. This will involve the reconfiguration of the model to predict pixel-wise segmentation masks that delineate texture boundaries, as opposed to inpainting missing regions. Training data for fine-tuning can be sourced either from labeled texture segmentation datasets (such as Real-World Textures [27, 28]) or even classic image segmentation datasets. In cases where semantic context appears, it is possible to extract from each image a set of proper sub-images that disregard the original semantics or randomly disassemble and reassemble the whole image (and the labels, correspondingly) in different orientations for the same purpose of achieving textural semantic-less set. The adapted model will then undergo testing on the TBM dataset, with performance evaluation against ground truth labels provided by domain experts. This comprehensive approach will result in a versatile UTBD capable of accurately detecting texture boundaries across various domains and applications.

Acknowledgments

This work was supported by the Israeli Ministry of Defence (MAFAT), the Pazy Foundation, and Intel Corporation (oneAPI Center of Excellence program). Computational support was provided by the NegevHPC project [1]. The authors would like to thank Gabi Dadush, Israel Hen, and Emil Malka for their hardware support on NegevHPC.

References

[1] NegevHPC Project. `www.negevhpc.com`. [Online].

[2] PV Andrews, MB West, and CR Robeson. The effect of grain boundaries on the electrical resistivity of polycrystalline copper and aluminium. *Philosophical Magazine*, 19(161):887–898, 1969.

[3] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

[4] RW Armstrong. The influence of polycrystal grain size on several mechanical properties of materials. *Metallurgical and Materials Transactions B*, 1(5):1169–1176, 1970.

[5] Arturo Barba-Pingarrón and Rafael González-Parra. Metallography and crystallographic texture analysis. *The Encyclopedia of Archaeological Sciences*, pages 1–4, 2018.

[6] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.

[7] Julesz Bela. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981.

[8] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4380–4389, 2015.

[9] Jeremy S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, Los Angeles, CA, USA, August 3-8, 1997*, pages 361–368, 1997.

[10] Gabriella Castellano, Leonardo Bonilha, LM Li, and Fernando Cendes. Texture analysis of medical images. *Clinical radiology*, 59(12):1061–1069, 2004.

[11] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[12] Brian L DeCost, Toby Francis, and Elizabeth A Holm. Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures. *Acta Materialia*, 133:30–40, 2017.

[13] Brian L DeCost, Matthew D Hecht, Toby Francis, Bryan A Webler, Yoosuf N Picard, and Elizabeth A Holm. Uhcsdb: Ultrahigh carbon steel micrograph database. *Integrating Materials and Manufacturing Innovation*, 6(2):197–205, 2017.

[14] Santa Di Cataldo and Elisa Ficarra. Mining textural knowledge in biological images: Applications, methods and trends. *Computational and structural biotechnology journal*, 15:56–67, 2017.

[15] Piotr Dollár and C. Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 06 2014.

[16] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[17] Anna Frühstück, Ibraheem Alhashim, and Peter Wonka. Tilegan: Synthesis of large-scale non-homogeneous textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–11, 2019.

[18] Leon Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*, pages 262–270, 2015.

[19] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bdcn: Bi-directional cascade network for perceptual edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):100–113, 2022.

[20] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 229–238, New York, NY, USA, 1995. ACM.

[21] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, et al. Segment anything model for medical images? *arXiv preprint arXiv:2304.14660*, 2023.

[22] Oleg Ivanov, Oxana Maradudina, and Roman Lyubushkin. Grain size effect on electrical resistivity of bulk nanograined bi2te3 material. *Materials Characterization*, 99:175–179, 2015.

[23] Wei Ji, Jingjing Li, Qi Bi, Wenbo Li, and Li Cheng. Segment anything is not always perfect: An investigation of sam on different real-world applications. *arXiv preprint arXiv:2304.05750*, 2023.

[24] Junfeng Jing, Shenjuan Liu, Gang Wang, Weichuan Zhang, and Changming Sun. Recent advances on image edge detection: A comprehensive review. *Neurocomputing*, 2022.

[25] B Julesz, E N Gilbert, L A Shepp, and H L Frisch. Inability of humans to discriminate between visual textures that agree in second-order statistics?revisited. *Perception*, 2(4):391–405, 1973.

[26] Bela Julesz. Visual pattern discrimination. *Information Theory, IRE Transactions on*, 8:84 – 92, 03 1962.

[27] Naeemullah Khan, Marei Algarni, Anthony Yezzi, and Ganesh Sundaramoorthi. Shape-tailored local descriptors and their application to segmentation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3890–3899, 2015.

[28] Naeemullah Khan and Ganesh Sundaramoorthi. Learned shape-tailored descriptors for segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–674, 2018.

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[30] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv: Computer Vision and Pattern Recognition*, 2015.

[31] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (ToG)*, 24(3):795–802, 2005.

[32] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[33] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10758–10768, 2022.

[34] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified texture synthesis with feed-forward networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 266–274, 07 2017.

[35] Gang Liu, Yann Gousseau, and Gui-Song Xia. Texture synthesis through convolutional neural networks and spectrum constraints. In *23rd International Conference on Pattern Recognition (ICPR)*, pages 3234–3239. IEEE, 2016.

[36] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001.

[37] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[38] Meysam Naghizadeh and Hamed Mirzadeh. Effects of grain size on mechanical properties and work-hardening behavior of aisi 304 austenitic stainless steel. *steel research international*, 90(10):1900153, 2019.

[39] Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40(1):49–70, 2000.

[40] Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. Edter: Edge detection with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1402–1412, June 2022.

[41] Matan Rusanovsky, Ofer Beeri, and Gal Oren. An end-to-end computer vision methodology for quantitative metallography. *Scientific Reports*, 12(1):1–27, 2022.

[42] Omry Sendik and Daniel Cohen-Or. Deep correlations for texture synthesis. *ACM Transactions on Graphics (TOG)*, 36(5):161, 2017.

[43] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3982–3991, 2015.

[44] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[45] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1349–1357, 2016.

[46] Ning Wang, Zhirui Wang, KT Aust, and Uwe Erb. Effect of grain size on mechanical properties of nanocrystalline materials. *Acta Metallurgica et Materialia*, 43(2):519–528, 1995.

[47] Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. *arXiv preprint arXiv:2304.03283*, 2023.

[48] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007.

[49] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[50] Hong Zeng, Ying Wu, Jiuxing Zhang, Chunjiang Kuang, Ming Yue, and Shaoxiong Zhou. Grain size-dependent electrical resistivity of bulk nanocrystalline gd metals. *Progress in Natural Science: Materials International*, 23(1):18–22, 2013.

[51] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Transactions on Graphics (TOG)*, 37(4), 2018.