Avoiding Post-Processing with Context: Texture Boundary Detection in Metallography

Inbal Cohen
Tel Aviv University
Israel
inbalc2@mail.tau.ac.il

Julien Robitaille
Clemex Technologies
Canada
julienr@clemex.com

Francis Quintal Lauzon
Clemex Technologies
Canada
francis@clemex.com

Ofer Beeri
IAEC
Israel
oferb@iaec.gov.il

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

Gal Oren
Stanford University, Technion
United States
galoren@stanford.edu

Abstract

Accurately identifying grain boundaries in metallographic images is challenging due to the intricate nature of texture boundaries. State-of-the-art (SOTA) models, like the Segment Anything Model (SAM), often fail in purely texture-based segmentation tasks without clear object boundaries. The specific case of models like SAM also requires prompts which in this context requires prior knowledge of grain position so the model can be seeded. Moreover, manual annotation is not only time-consuming but also subjective and context-sensitive. Current SOTA methods rely on small annotated patches for training and require extensive post-processing during inference to merge patch boundary maps. This approach often leads to overfitting to the ground truth and results in models that are not well-generalized.

We introduce MLOGRAPHY++, a novel approach that eliminates the need for post-processing by training on partially labeled context windows. Our method leverages a U-Net architecture trained with large context windows, where only a small portion is annotated, allowing the model to learn boundary segmentation in context. During inference, our model effectively handles partial and incomplete boundaries while accommodating context variations without the need for post-processing. To evaluate our approach, we adopt the Heyn intercept method, a classical technique for measuring average grain size, as a more suitable metric than pixel accuracy, and apply it to MLOGRAPHY++ and a fine-tuned AutoSAM model. This method better captures the critical distribution of grain sizes, which is difficult to label accurately on a pixel level. We benchmark MLOGRAPHY++ against the SOTA MLOgraphy [20] on the Texture Boundary in Metallography (TBM) dataset [21]. Our results demonstrate that MLOGRAPHY++ achieves comparable performance while eliminating the need for post-processing, thus enhancing the generalizability of the method. This work highlights the importance of contextual training in improving the accuracy and practicality of texture boundary detection in metallography. [1]

1 Introduction

Material science and Quantitative Metallography (QM) [6] are pivotal disciplines in understanding and optimizing the properties of materials by analyzing their microstructural features. These fields

[1] Source code and dataset: https://github.com/Scientific-Computing-Lab/MLOgraphyPlusPlus.

AI for Accelerated Materials Discovery (AI4Mat) Workshop, NeurIPS 2024.

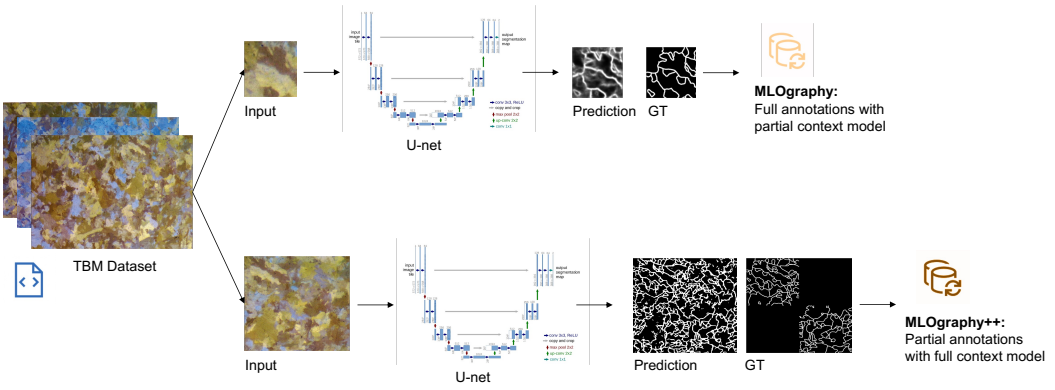


Figure 1: **Small annotated patch versus large partially annotated context window training:** Top: Previous methods, such as MLOgraphy, assume full annotation of a partial context window. They train on small 128x128, fully annotated image patches. The U-Net learns to predict the edge map using a fully annotated GT edge map. Bottom: Our method, MLOGRAPHY++, on the other hand, uses partial annotations with a full context window. That is, we use partial labels while preserving the entire metallographic scan image as a background. The U-Net learns to predict the entire context window, given just a partially annotated edge map as the GT to compare against. The images are tiled to 320x320 pixels, enabling the model to capture full contextual information during training.

investigate how material properties are influenced by chemical composition, microstructure, and manufacturing processes [24]. QM [20, 6], a specialized area within material science, focuses on the microstructural analysis of materials, particularly metallic alloys, at scales ranging from nanometers to millimeters.

Grain texture [29] is a fundamental microstructural feature in materials that significantly influences their properties. Grains are regions within a material where atoms are arranged in specific crystallographic orientations. These grains form during recrystallization or phase transitions like solidification. The spatial distribution of grains, influenced by different growing kinetics and mechanical shaping, affects material properties like yield strength, ductility, and toughness. The interfaces between adjacent grains, known as grain boundaries, are critical features that impact mechanical properties and are often the focus of detailed studies [20, 3, 13] to understand and improve material performance.

Automating the analysis of grain boundaries detection presents several significant challenges:

1. Textural Transitions: Grains represent textural transitions that depend on relationships within and between different material models. This makes locating grain boundaries challenging due to gradual transitions, varying relationships, and differing textural characteristics across models.
2. Contextual Dependence: Microstructural features can vary based on imaging conditions and surrounding elements. This variability can lead to inconsistent segmentation outcomes for the same image, highlighting the critical role of context in accurate prediction.
3. Ambiguity in Boundaries: Grain boundaries may not always form clear, closed contours within the dataset. Partial or incomplete boundaries create ambiguity, complicating the segmentation process and leading to inconsistent results.

These challenges are well-illustrated in [21], which showed that even strong models, such as EDTER [18], which is a transformer-based edge detection model, struggle with texture issues during segmentation tasks, as seen in Texture Boundary in Metallography comprehensive dataset (henceforth, TBM dataset [21]) and others [5, 11]. The study highlighted how performance metrics such as fixed contour threshold (ODS), per-image best threshold (OIS), and average precision (AP) were significantly lower when the model lacked complete image context. Additionally, the segmentation results for grain boundaries were notably better when an enlarged image context was provided.

This observation aligns with findings from other research, particularly [20], which emphasized that many existing segmentation models, specifically MLOgraphy, struggle due to the small and partial images often found in datasets like the TBM dataset [20]. The lack of sufficient contextual

information in these images can significantly impair the accuracy of these models, leading to increased noise, incomplete predictions, and reduced reliability in grain boundary detection. These findings underscore the critical importance of contextual information in texture boundary detection, which is a key focus of our current research.

These challenges are particularly addressed by the Texture Boundary in Metallography (TBM) dataset, which serves as the primary dataset for this work. The TBM dataset was specifically designed to highlight the complexities encountered in metallographic image analysis, offering high-resolution images with intricate texture transitions and partial boundaries. It effectively captures the wide variety of grain structures and boundary ambiguities present in real-world metallographic images, making it an ideal benchmark for evaluating models that must generalize well to these complexities. By using the TBM dataset, we ensure that our model confronts real-world challenges in grain boundary detection, offering a reliable testbed for methods aimed at improving segmentation accuracy and contextual understanding.

Contributions. We introduce a U-Net texture boundary detection model with partial labeling named MLOGRAPHY++ to overcome the limitations of previous methods, particularly in comparison to the SOTA MLOgraphy [20] model. Unlike earlier approaches that assumed complete grain boundaries [20, 31, 15], MLOGRAPHY++ effectively handles partial and incomplete boundaries while accommodating context variations. By utilizing partial labels to identify regions as grain edges or backgrounds, our model prioritizes continuous edge detection, providing a more accurate and flexible solution for grain boundary detection and segmentation. We demonstrate that MLOGRAPHY++ matches ground truth (GT) and MLOgraphy predictions without the need for post-processing. Figure 1 illustrates the difference between MLOGRAPHY++ and MLOgraphy.

Additionally, we employ a variation of the computerized Heyn intercept method (ASTM E112) [14], termed the Heyn-Compare method, as the relevant segmentation evaluation metric for this problem, suitable for determining average grain size accurately, even with incomplete grain boundaries. This variation of the Heyn intercept method offers more precise evaluations than common pixel accuracy metrics like IoU and Dice coefficient. Unlike these metrics, which often struggle with incomplete boundaries and context variations, our method provides quantitative, context-aware assessments that are robust to image quality variations and directly applicable in material science. This ensures more relevant and reliable evaluations from both machine-learning and physical perspectives.

The rest of the paper is organized as follows: In section 2, we provide an overview of related work, discussing existing methods and their limitations in the context of grain boundary detection. section 3 details the limitations of the current SOTA method, MLOgraphy. In section 4, we introduce our proposed method, MLOGRAPHY++, explaining its architecture, training process, and inference mechanism. Moreover, section 5 describes our evaluation methodology, including the adoption of the Heyn intercept method and the metrics used for comparison, while in section 6 we also apply it to AutoSAM fine-tuned results. Finally, section 7 discusses the limitations of our work and potential future directions.

2 Related Work

Grain boundary segmentation faces many challenges due to the frequent absence of complete grain boundaries, the complicating texture transitions, and the contextual dependence. These issues lead to ambiguity in model training and evaluation.

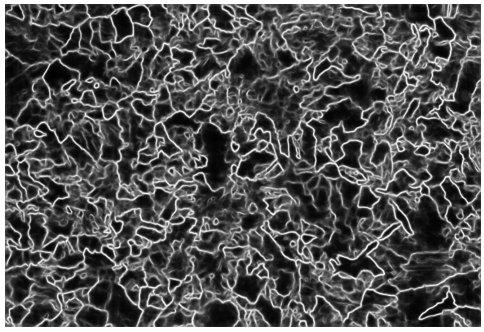
Deep learning techniques have revolutionized grain analysis. U-Net, originally designed for biomedical image segmentation [19], has been adapted for metallography, significantly improving grain boundary detection by capturing fine details in complex images. However, even SOTA methods like MLOgraphy [20], which uses a U-Net architecture, face challenges. Trained on cropped sub-images with expert annotations but lacking full contextual information, MLOgraphy often produces inconsistent and fragmented boundary predictions. These incomplete predictions, interpreted as noise, ultimately reduce the accuracy and reliability of the segmentation, necessitating additional post-processing steps to suppress the noise and compensate for the missing contextual information.

The recent Segment Anything Model (SAM) [9], leveraging a transformer architecture, has shown improvements in various image analysis tasks. However, it struggles with pure texture images and often fails at accurate segmentation without clear object boundaries. As mentioned in [7],

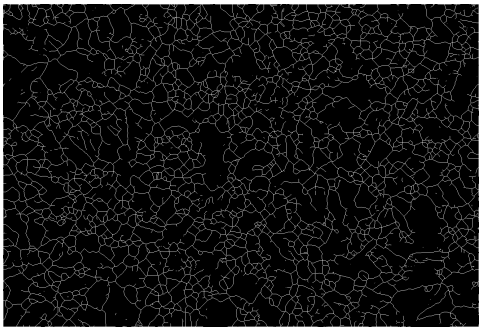
SAM faces significant challenges in handling complex scenes, low-contrast objects, and smaller or irregular objects, as seen in metallography. Similarly, MedSAM [12], designed for universal medical image segmentation, or MicroSam [4], designed for cell segmentation, also struggle with complex textures and unclear boundaries. Moreover, SAM's versatility comes also at the cost of being resource-intensive, making it less ideal for specialized, real-time metallographic applications.

The TBM dataset [21] highlights the challenges of segmenting metallographic images with complex textures and incomplete boundaries. To address these, in this work, we trained U-Net with partial labels on the TBM dataset while keeping the complete context of the image. This approach, as will be further explained, shows improved accuracy and reliability over previous methods (especially MLOgraphy), demonstrating robustness in tackling TBM's unique challenges.

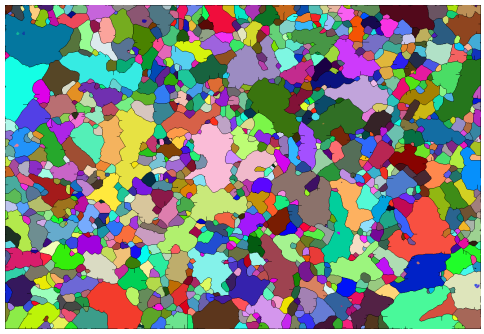
3 MLOgraphy's Limitations



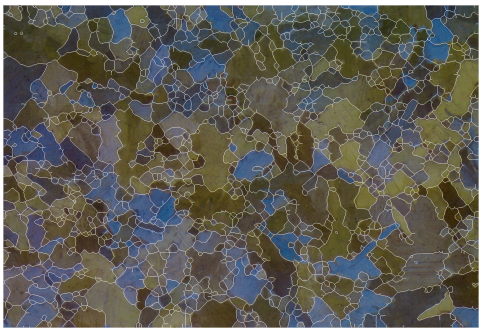
(a) Full raw GB segmentation.



(b) Binarized raw GB segmentation.



(c) Watershed contours after binarization.



(d) Full GB segmentation.

Figure 2: **Post-processing limitations in MLOgraphy** [20]: The figure illustrates the post-processing workflow applied by MLOgraphy to the raw output of the model. Initially, the model's predictions exhibit noise due to variance in boundary predictions, resulting in incomplete lines (a). To address this, binarization followed by Guo-Hall thinning is performed (b). The Watershed algorithm is then applied to eliminate the incomplete lines, retaining only 'certain' boundaries. The final contours generated from this process are shown in (c), with the post-processed boundaries overlaid on the input image in (d).

While MLOgraphy provides a promising approach for grain boundary detection, it suffers from several limitations. The model's output, which predicts boundaries at the pixel level, often exhibits variability in predicted values across different boundaries and even along the same boundary line. This inconsistency results in incomplete or fragmented boundary lines, which reflect the model's uncertainty. Such fragmented lines are typically treated as noise, leading to the potential loss of critical boundary information.

Another significant limitation arises from the necessity of post-processing steps (see Figure 2) to mitigate the noise and enhance the clarity of boundaries. The application of binarization and the

Watershed algorithm is essential for eliminating incomplete boundaries and ensuring that only the 'certain' boundaries are retained. However, they also highlight the model's dependency on external processes to produce reliable and interpretable results.

In summary, while MLOgraphy offers a systematic and potentially time-saving method for grain boundary detection, it faces challenges in consistency, the need for extensive post-processing, and maintaining uniformity in results. In this paper, we introduce MLOGRAPHY++, a new method specifically designed to address and solve some of these limitations.

4 MLOGRAPHY++: Texture Boundary Detection Using Partial Labeling

Our new method, MLOGRAPHY++, aims to identify grain boundaries by focusing on one-dimensional objects, reducing both false negatives and positives without requiring complete grain contours. MLOGRAPHY++ utilizes a U-Net architecture trained with partial labels to better address context and texture issues. Labels indicate regions as either part of the grain edge (foreground) or not (background), prioritizing the detection of continuous edges rather than complete contours. This approach enables more accurate identification of grain boundaries, addressing the inherent ambiguities in metallographic images and their strong contextual meaning (section 5). We henceforth provide a comparison of the training and inference processes for both MLOgraphy and MLOGRAPHY++, highlighting key differences (see Figure 1 and Figure 3, with system demonstration at Appendix B).

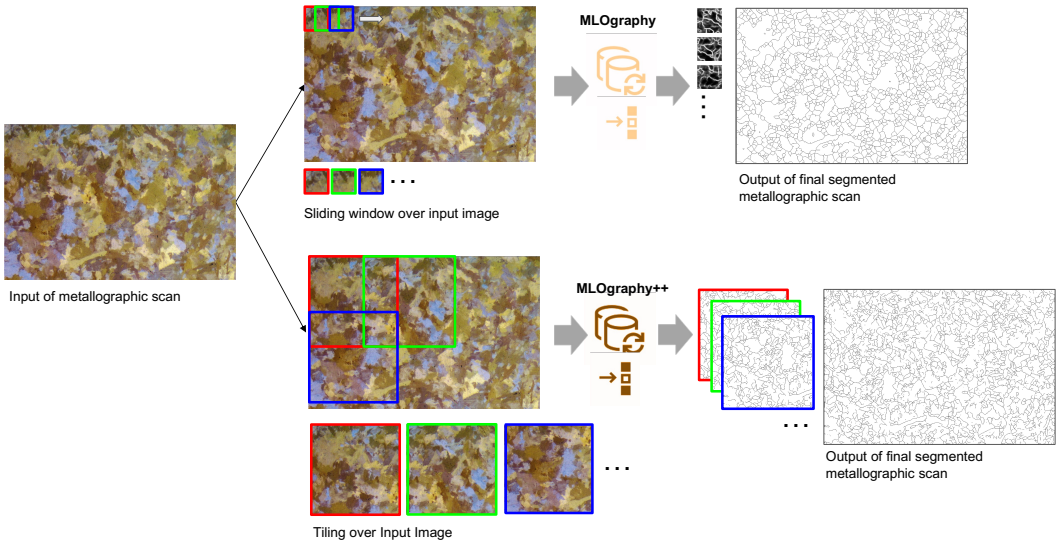


Figure 3: **Inference illustration comparing MLOgraphy and MLOGRAPHY++** using an example image from the TBM dataset. MLOgraphy uses a sliding window to create overlapping 128x128 crops of the input metallographic scan, which are then processed individually by the U-Net model. These crops undergo post-processing, including binarization, Guo-Hall thinning, and the Watershed algorithm, to refine boundaries and generate the final segmented scan. In contrast, the new method, MLOGRAPHY++, processes the entire image directly(tiled to 320x320 pixels), using only Guo-Hall thinning, to produce the final segmented scan without additional post-processing steps. Guo-Hall thinning is applied at the end of the evaluation to generate thin boundary lines for analysis. The results are accurate enough before thinning, so no additional post-processing, like binarization or the Watershed algorithm, is needed.

- **Training**: Both methods leverage the TBM dataset with identical labeling and a U-Net architecture; however, the primary distinction lies in their training approaches (see Figure 1). MLOGRAPHY++ uses partial labels while preserving the entire metallographic scan image as background. This is accomplished by calculating the loss only on the annotated pixels rather than the entire image. The U-Net model used is of depth 3 and is adapted with a MobileNetV2 [22] encoder. Training is conducted using the ADAM optimizer, with pre-trained weights initialized from MicroNet [25]. To address the class imbalance, the Cross-Entropy loss function's weights are calculated using median

frequency balancing [8][1]. Images are tiled to 320x320 pixels, following the approach detailed by Possolo and Bajcsy [17]. The tiling process allows for the training of larger images by dividing them into smaller tiles that fit within GPU memory, ensuring accurate results. Key concepts include the Zone of Responsibility (ZoR), which is the area being processed; the Halo, a border providing necessary context for accurate computation; and the Stride, the step size used to create tiles. In our implementation, the ZoR is 320x320 pixels, and the Halo is 96 pixels.

- **Inference**: For inference (see Figure 3), MLOgraphy uses a high-overlapping sliding-window methodology, creating overlapping 128x128 crops, which are processed individually and combined using a majority vote method. In contrast, MLOGRAPHY++ processes the entire image (tiled to 320x320 pixels, following the approach detailed in [17]), resulting in improved segmentation accuracy and efficiency by capturing the needed image context.

5 Evaluation of Grain Boundary Detection using Heyn Method Variation

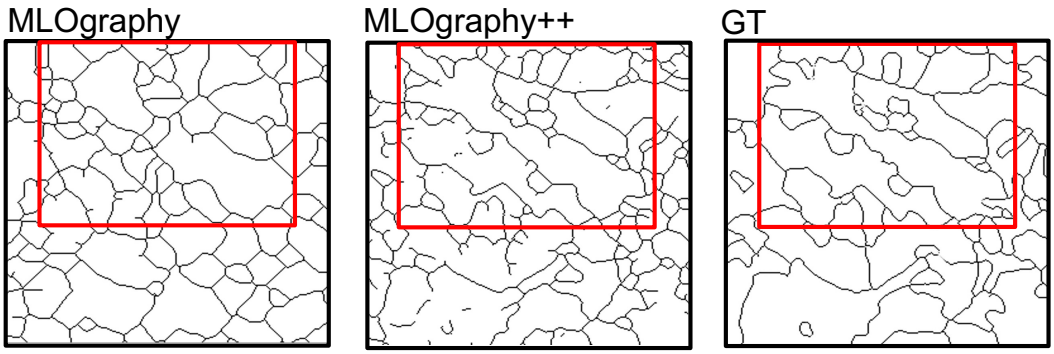


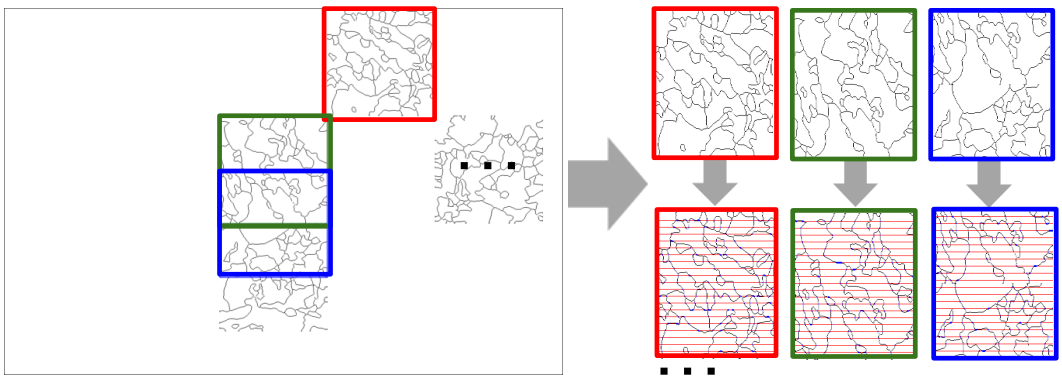
Figure 4: **IoU Limitations as an evaluation method for texture boundary detection in metallography**: The IoU was calculated over a zone of a specific grain (highlighted in a red window) in MLOgraphy, GT, and MLOGRAPHY++ samples. MLOgraphy achieved an IoU score of 0.0686, while MLOGRAPHY++ reached 0.2180. These low IoU values highlight the inadequacy of IoU as an evaluation metric for texture boundary segmentation, as it fails to accurately capture grain boundaries.

In previous methods, grain boundaries were evaluated based on pixel similarity metrics, such as IoU and Dice coefficient, which often led to inaccuracies due to the assumption that boundaries are always fully visible in the image [30, 16, 2]. However, grain boundaries, while continuous in reality, are not always discernible in metallographic images, causing segmentation ambiguities. Early methods like the Jeffries Planimetric [28] and Triple-Point Count [26] involved manual counting within defined areas, which, although systematic, were time-consuming and prone to errors, especially when boundaries were unclear or grain structures irregular.

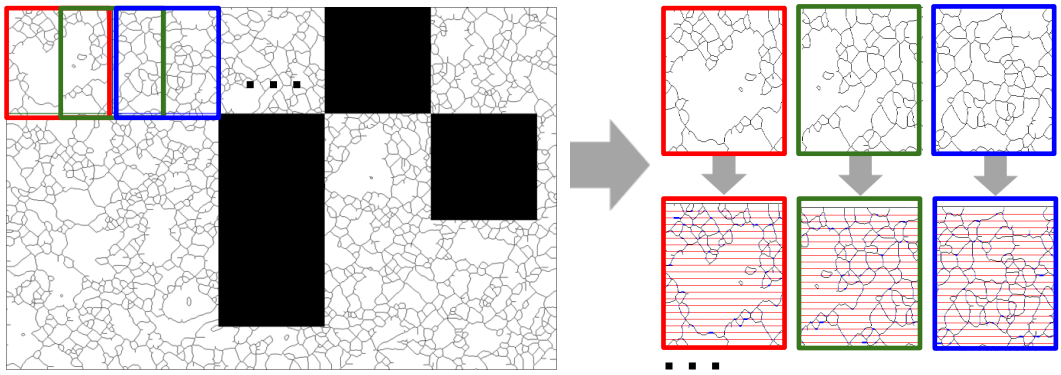
In Figure 4, we present the Intersection over Union (IoU) scores for the segmentation results of MLOgraphy and MLOGRAPHY++ for grain boundaries, highlighting the limitations of IoU in texture boundary segmentation. The low IoU scores underscore the inadequacy of this metric for detecting grain boundaries. Since IoU assumes fully enclosed boundaries and penalizes small deviations, it is not well-suited for grain boundaries, which are typically continuous and fragmented. In both MLOgraphy and MLOgraphy++ segmentations, the grain boundaries are not always fully closed, underscoring the unsuitability of IoU for this task. By focusing on boundary intersections rather than pixel accuracy, our variation of the Heyn intercept method provides a more suitable evaluation, particularly for handling partial and irregular boundaries, which are common in metallographic images.

The standard Heyn method involves placing lines of known length across a micrograph and counting how often these lines intersect with grain boundaries. This can be done using straight or circular test lines, depending on the grain structure being analyzed. The average grain intercept (AGI), a measure of the average grain size, is then calculated using the formula: $\text{AGI} = \frac{\text{Number of Intercepts}}{\text{Total Line Length}}$.

[1]This approach, as proposed in the referenced paper, proved effective in our application.



(a) Heyn-Compare method applied to GT crops.



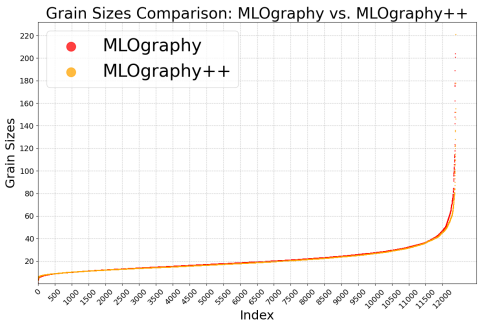
(b) Heyn-Compare method applied to full predictions without GT crops.

Figure 5: **Evaluation of grain boundary detection using the Heyn-Compare method**: The process compares MLOgraphy and MLOGRAPHY++ trained with partial labels against the GT. We extracted 256x256 image crops per model, avoiding overlap with the GT, and included crops that overlap by 50% with each other. A structured pattern of 20 horizontal lines was applied for each crop for evaluation. The figure illustrates how the crops appear after applying the Heyn-Compare method and demonstrates the overall process.

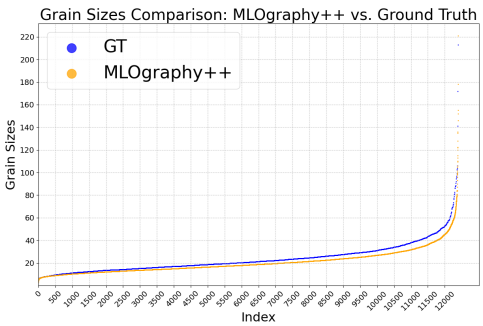
However, in our study, we opted for a variation of this method to address specific challenges associated with grain boundary detection in metallographic images, as discussed by [27]. While the traditional Heyn method is effective, it assumes that grain boundaries are well-defined and fully visible, which is not always the case in practice. In metallographic images, boundaries can be partially obscured or poorly resolved due to factors such as deformation, anisotropy, or imaging limitations. Additionally, when using circular test grids, the curvature of the lines can introduce biases and complexities in counting intercepts, potentially leading to underestimation of grain sizes, particularly when small circles are used. These challenges can result in inaccuracies in grain size measurements, especially when the boundaries are not straightforward to identify or when the test lines do not align optimally with the grain structures.

To address these issues, we use a variation of the Heyn intercept method (ASTM E112) [10, 14] termed **Heyn-Compare**, which measures average grain size without requiring closed contours, providing a more robust evaluation. While cross-entropy is used as the loss function during training for its effectiveness in pixel-wise classification, allowing the model to accurately segment grain boundaries, our variation of the Heyn intercept method is employed as the measure of success because it directly assesses grain size, the primary objective of our analysis. This approach ensures the model is both effectively trained and evaluated on a metric aligned with our study's practical goals.

In the Heyn-Compare method, we draw fixed horizontal lines across the sample and measure the distances between adjacent points where these lines intersect with grain boundaries. Instead of merely



(a) Comparison of MLOgraphy and MLOGRAPHY++. MLOGRAPHY++ closely aligns with MLOgraphy without the need for post-processing. This is achieved by an enlarged model context.



(b) Comparison of MLOGRAPHY++ with GT. MLOGRAPHY++ closely matches the GT, providing accurate and reliable predictions more efficiently, without requiring post-processing.

Figure 6: **Heyn-Compare Comparison** of grain sizes predicted by MLOgraphy, MLOGRAPHY++, and GT. The grains are ordered by their grain size. Except few outliers at the end of the scale (indicating exceptionally large grain sizes) most of the grain sizes that have been measured are in a closely matching pattern.

counting the intersections, we calculate the average distance between these adjacent points along the grain boundaries. The average grain size for a given line L_i, denoted as $\bar{D}_i$, is calculated as $\bar{D}_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i - 1} d_{ij}$, where $d_{ij} = \|P_{i(j+1)} - P_{ij}\|$ represents the Euclidean distance between two adjacent intersection points P_{ij} and $P_{i(j+1)}$. The global average grain size for the entire image is then calculated by averaging the grain sizes from all n lines as $\bar{D} = \frac{1}{n} \sum_{i=1}^{n} \bar{D}_i$.

This approach provides a more precise estimate of grain size across the entire image. It is particularly effective for evaluating both complete and partial grain boundaries, offering a consistent metric across different image scales and resolutions. When applied with sufficient GT labels, this method allows for reliable comparisons of segmentation techniques by analyzing both the average grain size and the variance in grain sizes within sub-regions of metallographic image predictions, thereby enhancing the robustness and reliability of the analysis.

Using the Heyn-Compare method on the TBM dataset [21], we plotted two graphs comparing the grain sizes from human annotations (GT) with the grain sizes predicted by the two models (see Figure 6, while Figure 8 in Appendix A shows a breakdown for each sample, which mostly follows the general trend from Figure 6). These graphs allow for a direct comparison of grain size distributions produced by different segmentation methods against the GT. The graphs show the grain sizes for each sample, sorted in ascending order. As can be seen, MLOGRAPHY++ closely aligns with MLOgraphy without the need for post-processing. This is achieved by an enlarged model context.

To maintain consistency, we used fixed lines instead of random ones, which is reasonable given the equiaxed nature of the grains in our dataset. From the TBM dataset, which includes 21 models with partial labels, we extracted all non-overlapping 256x256 image crops per model, as well as crops that overlap by 50% with each other. For the GT, we also cropped 256x256 images with a 50% overlap with each other. We applied a structured pattern of 20 horizontal lines at fixed locations to these images, reducing variability and ensuring reliable evaluation (see Figure 5). Since the comparison of mean and variance across models was inconsistent, we focused on the overall grain size distribution. After obtaining the grain sizes, we then sorted them in ascending order to facilitate a clear comparison across models.

This sorting helped us identify outliers, which may result from abnormal grain growth, defects, or mechanical deformation, significantly impacting material properties and enabling a more accurate comparison across models. Outliers in grain size distributions are typically defined as grains that deviate significantly from the mean, often identified through statistical methods like standard deviation or interquartile range. These outliers can influence material properties—larger grains may reduce strength and toughness, while smaller grains can enhance hardness but may cause embrittlement. Recognizing and accounting for these outliers allows for a more accurate assessment of material performance.

The results show that both MLOgraphy and MLOGRAPHY++ closely align with the GT. However, a key distinction is that MLOGRAPHY++ does so without requiring additional post-processing, making it more generalizable and efficient and highlighting the contribution of context to efficient and effective segmentation in metallographic data.

6 Comparative Evaluation of AutoSAM in Metallographic Segmentation

SAM, though highly effective in general segmentation tasks, struggles with the intricate grain boundaries and subtle phase variations inherent in metallographic images. SAM, originally trained on large-scale natural image datasets, often fails to capture the nuances of metallographic textures, making it difficult to segment grain boundaries accurately and distinguish between similar phases.

To address these challenges, we fine-tuned AutoSAM [23] on the TBM (Texture Boundary in Metallography) dataset. Unlike SAM, which relies on prompts, AutoSAM replaces this dependency with an encoder that processes input images directly, enabling fully automated segmentation. By leveraging gradients from a frozen SAM model, AutoSAM has shown promise in other out-of-distribution domains, including medical imaging.

We trained AutoSAM using a method similar to the MLOgraphy approach, fine-tuning it on 256x256 fully annotated crops from the TBM dataset for 100 epochs. The training process used a batch size of 2 and was optimized with a combination of Binary Cross-Entropy Loss (BCELoss) and Dice Loss. The Adam optimizer was applied with a learning rate of 0.0003 and a weight decay of 0.0001, mirroring the configuration used in AutoSAM's original settings.

To improve robustness and adaptability to complex grain structures, we applied several data augmentation techniques: color jitter (brightness, contrast, and saturation set to 0.4, and hue to 0.1), random vertical and horizontal flips, and random affine transformations (maximum rotation of 90 degrees and scaling factors between 0.75 and 1.25). After these augmentations, the training set consisted of 128 images and the test set of 16 images. These augmentations were crucial for handling the complex grain boundaries and phase variations in metallographic images.

After training, we applied a series of post-processing steps, similar to the MLOgraphy method, to refine the predicted masks. These steps included Otsu's thresholding to binarize both the predicted and ground truth masks, followed by the watershed algorithm to better segment complex grain structures and separate touching objects. Finally, we used the Guo-Hall thinning algorithm to refine the boundary structures, enhancing the model's ability to capture intricate grain patterns.

After applying post-processing steps to both the ground truth and predicted masks on the test set, we observed low values: a Mean IoU of 0.1188 and a Mean Dice Score of 0.2121. These results suggest that while IoU and Dice are commonly used for evaluating segmentation tasks, they may not be entirely suitable in this context. The complex mask structures in metallography lead to unusually low metric values that do not fully reflect the model's performance, as discussed in section 5. Despite this, the visual comparison in Figure 7 highlights that AutoSAM's predictions capture intricate grain boundary structures that the metrics fail to account for.

In contrast, when using the Heyn-Compare method, AutoSAM's performance closely matched the ground truth (GT) grain sizes. This method, which evaluates boundary intersections rather than pixel-wise accuracy, provided a more reliable measure of grain size, particularly when dealing with partial and irregular grain boundaries. As shown in Table 1, AutoSAM's grain size predictions were closely aligned with the GT values, demonstrating the model's ability to estimate grain sizes accurately even in complex metallographic images. These results confirm that the Heyn-Compare method is a more appropriate metric for this task.

Mean Grain Size AutoSAM	25.27
Mean Grain Size GT	24.842
Mean IoU	0.1188
Mean Dice	0.2121

Table 1: Summary of Mean Grain Size, IoU, and Dice scores for AutoSAM and Ground Truth (GT).

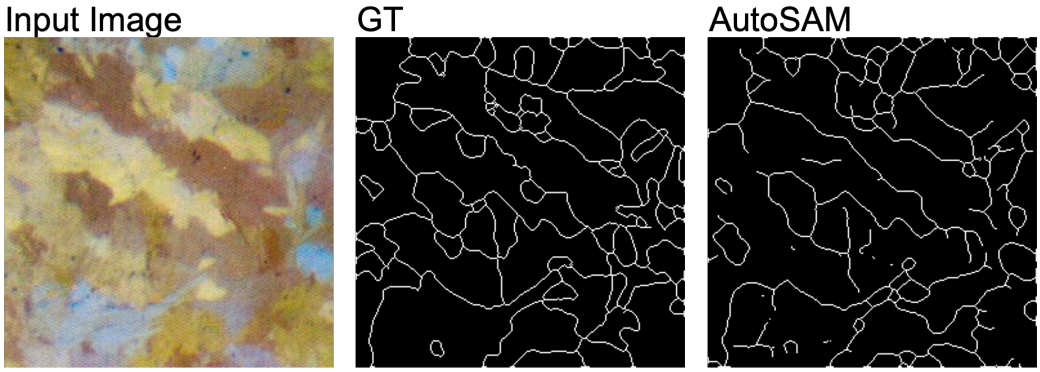


Figure 7: Comparison of Input Image, Ground Truth (GT), and AutoSAM Predictions

7 Conclusions, Limitations and Future Work

In this paper, we introduced MLOGRAPHY++, an innovative approach for texture boundary detection in metallography that leverages partial labeling and enhanced contextual training. Our method addresses key limitations of existing techniques, such as MLOgraphy, which rely heavily on fully annotated small image patches and extensive post-processing. By utilizing partial labels and training on larger context windows, MLOGRAPHY++ effectively captures the intricate and often incomplete grain boundaries inherent in metallographic images. Evaluations using the Heyn-Compare method demonstrated that MLOGRAPHY++ not only matches but often surpasses the performance of existing methods, providing accurate grain size measurements without the need for post-processing. This advancement underscores the importance of contextual information in texture boundary detection and presents MLOGRAPHY++ as a valuable tool for material scientists and engineers engaged in microstructural analysis.

Despite its promising results, MLOGRAPHY++ has several limitations that warrant further investigation. The primary limitation is the current reliance on the TBM dataset for evaluation, which may not fully represent the diversity of microstructural variations in metallographic images. Additionally, the increased model complexity due to the use of larger context windows and partial labeling results in higher computational costs during training and inference. The dependence on partial labels, while beneficial for capturing incomplete boundaries, introduces potential biases if the labels are not consistently representative.

Future work will focus on expanding dataset coverage to include a wider variety of metallographic images, fine-tuning transformer-based models like SAM for specialized metallographic segmentation, and optimizing model efficiency. Furthermore, developing automated methods for generating high-quality partial labels will be explored to mitigate the dependency on manual labeling. By addressing these areas, we aim to enhance MLOGRAPHY++'s robustness, efficiency, and applicability, thereby advancing the field of quantitative metallography.

Acknowledgments

This work was supported by the Pazy Foundation. Computational support was provided by the NegevHPC project [1].

References

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

[2] Doruk Aksoy, Huolin L Xin, Timothy J Rupert, and William J Bowman. Human perception-inspired grain segmentation refinement using conditional random fields. *arXiv preprint arXiv:2312.09968*, 2023.

[3] 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.

[4] Anwai Archit, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Sagnik Gupta, Andreas Dengel, Sheraz Ahmed, and Constantin Pape. Segment anything for microscopy. *bioRxiv*, pages 2023–08, 2023.

[5] 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.

[6] Brian L DeCost, Bo Lei, Toby Francis, and Elizabeth A Holm. High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel. *Microscopy and Microanalysis*, 25(1):21–29, 2019.

[7] Wei Ji, Jingjing Li, Qi Bi, Tingwei Liu, Wenbo Li, and Li Cheng. Segment anything is not always perfect: An investigation of sam on different real-world applications, 2024.

[8] Junfeng Jing, Zhen Wang, Matthias Rätsch, and Huanhuan Zhang. Mobile-unet: An efficient convolutional neural network for fabric defect detection. *Textile Research Journal*, 92(1-2):30–42, 2022.

[9] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[10] Xiang Li, Linyi Cui, Jikang Li, Ying Chen, Wei Han, Sara Shonkwiler, and Sara McMains. Automation of intercept method for grain size measurement: A topological skeleton approach. *Materials & Design*, 224:111358, 2022.

[11] Julian Luengo, Raul Moreno, Ivan Sevillano, David Charte, Adrian Pelaez-Vegas, Marta Fernandez-Moreno, Pablo Mesejo, and Francisco Herrera. A tutorial on the segmentation of metallographic images: Taxonomy, new metaldam dataset, deep learning-based ensemble model, experimental analysis and challenges. *Information Fusion*, 78:232–253, 2022.

[12] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024.

[13] M Mohammadtaheri. A new metallographic technique for revealing grain boundaries in aluminum alloys. *Metallography, Microstructure, and Analysis*, 1:224–226, 2012.

[14] J Muirhead, J Cawley, A Strang, CA English, and J Titchmarsh. Quantitative aspects of grain size measurement. *Materials science and technology*, 16(10):1160–1166, 2000.

[15] Matthew J Patrick, James K Eckstein, Javier R Lopez, Silvia Toderas, Sarah A Asher, Sylvia I Whang, Stacey Levine, Jeffrey M Rickman, and Katayun Barmak. Automated grain boundary detection for bright-field transmission electron microscopy images via u-net. *Microscopy and Microanalysis*, 29(6):1968–1979, 2023.

[16] V. Pece, C. M. Gorsevski, J. R. Onasch, J. R. Farver, and X. Ye. Detecting grain boundaries in deformed rocks using a cellular automata approach. *Computers & Geosciences*, 42:136–142, 2012.

[17] Michael Possolo and Peter Bajcsy. Exact tile-based segmentation inference for images larger than gpu memory. *Journal of Research of the National Institute of Standards and Technology*, 126, 2021.

[18] 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*, pages 1402–1412, 2022.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[20] Matan Rusanovsky, Ofer Beeri, and Gal Oren. An end-to-end computer vision methodology for quantitative metallography. *Scientific Reports*, 12(1):4776, 2022.

[21] Matan Rusanovsky, Ofer Be'eri, Shai Avidan, and Gal Oren. Universal semantic-less texture boundary detection for microscopy (and metallography). In *NeurIPS 2023 Workshop on Machine Learning and the Physical Sciences*, 2023.

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[23] Tal Shaharabany, Aviad Dahan, Raja Giryes, and Lior Wolf. Autosam: Adapting sam to medical images by overloading the prompt encoder. *arXiv preprint arXiv:2306.06370*, 2023.

[24] Anil Kumar Sinha. Physical metallurgy handbook. *(No Title)*, 2003.

[25] J Stuckner, B Harder, and TM Smith. Microstructure segmentation with deep learning encoders pre-trained on a large microscopy dataset. npj comput. mat. 8 (1), 200, 2022.

[26] George F Vander Voort. Grain size measurements by the triple point count method. *Practical Metallography*, 51(3):201–207, 2014.

[27] GF Vander Voort. Examination of some grain size measurement problems. In *Metallography: Past, Present, and Future (75th Anniversary Volume)*. ASTM International, 1993.

[28] GF Vander Voort. Grain size measurements using circular or rectangular test grids. *Practical Metallography*, 50(1):17–31, 2013.

[29] Lifei Wang, Ehsan Mostaed, Xiaoqing Cao, Guangsheng Huang, Alberto Fabrizi, Franco Bonollo, Chengzhong Chi, and Maurizio Vedani. Effects of texture and grain size on mechanical properties of az80 magnesium alloys at lower temperatures. *Materials & Design*, 89:1–8, 2016.

[30] Yu Han Wang, Qing He, and Zhi Xie. Grain boundary extraction method based on pixel relationship. *Measurement*, 202:111796, 2022.

[31] Mallory Wittwer, Bernard Gaskey, and Matteo Seita. An automated and unbiased grain segmentation method based on directional reflectance microscopy. *Materials Characterization*, 174:110978, 2021.

A Appendix: Distributions breakdown for each sample

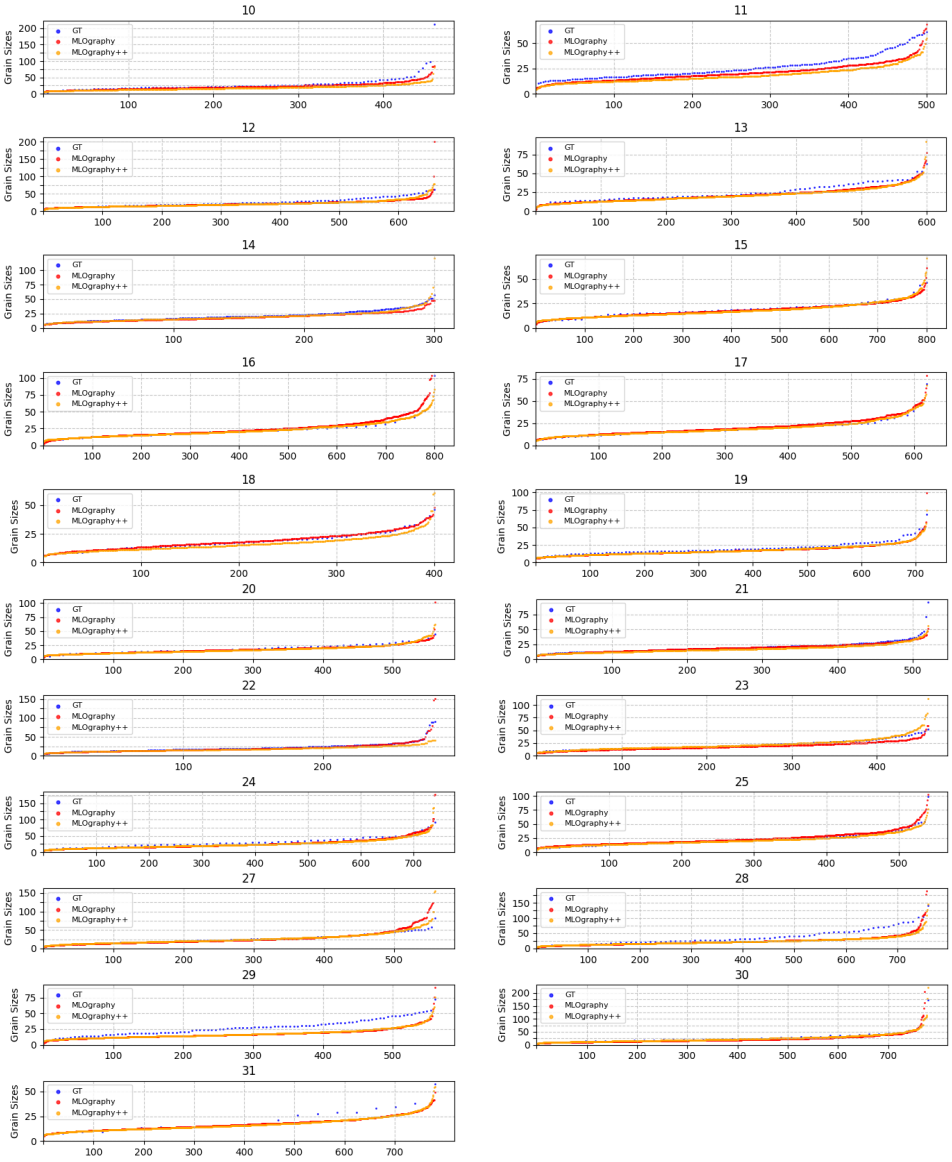
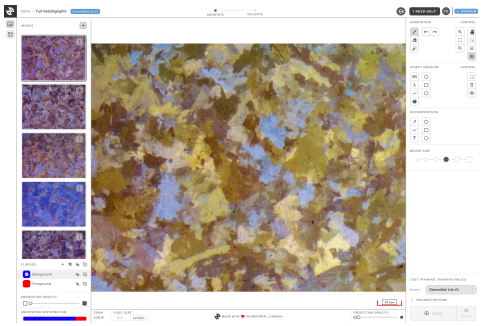


Figure 8: Direct Heyn-Compare comparison of grain size distributions breakdown for each sample (a breakdown of the general trends in Figure 6).

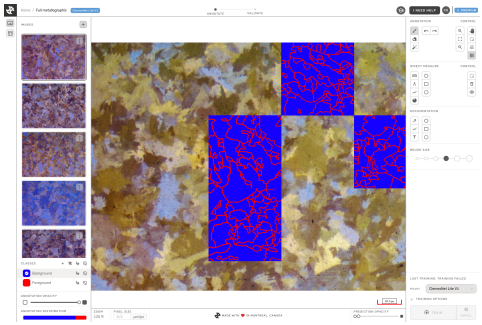
B Appendix: Clemex Studio with MLOGRAPHY++

Clemex Studio [2] is a no-code platform for developing semantic segmentation algorithms using partial annotations. It was made for domain experts without machine learning experience. This tool was utilized to train the MLOGRAPHY++ model through the ClemexNet Lite V1 algorithm, as described in section 4.

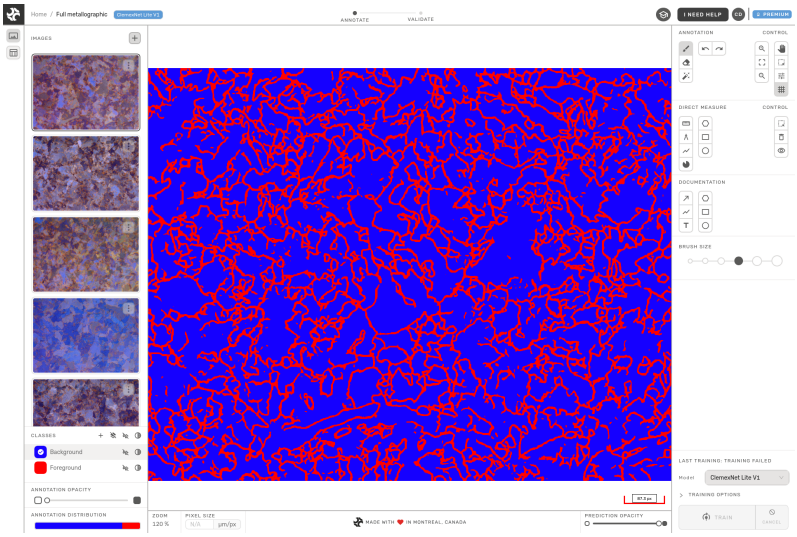
The platform provides a streamlined workflow, enabling users to upload images, annotate them, and train segmentation algorithms based on partial annotations.



(a) Initial image in Clemex Studio after upload.



(b) Example of an added annotation in Clemex Studio.



(c) The Clemex Studio training has been successfully completed, with prediction results demonstrated.

Figure 9: Illustration of the iterative workflow in Clemex Studio using MLOGRAPHY++ on the TBM dataset. (a) Shows the initial image upon upload, (b) demonstrates the addition of annotations to improve prediction accuracy, and (c) displays the final prediction results after training is completed.

Clemex Studio facilitates a novel approach to algorithm training through an iterative workflow. Users need only annotate a subset of a single image to initiate the training process, resulting in predictions for the entire image. During each iteration, users can refine the model by adding additional annotations to areas of misprediction, progressively enhancing the model's accuracy. To incorporate robust algorithm development practices, Clemex Studio also includes a validation page as part of the workflow for evaluating model predictions on previously unseen data.

Additionally, users can export segmentation masks as a zip file or save the model and selected algorithm as a plugin for integration with Clemex Vision.

[2] https://studio.clemex.ai