scientific reports

OPEN

An end-to-end computer vision methodology for quantitative metallography

Matan Rusanovsky[1,2], Ofer Beeri[3] & Gal Oren[1,4]✉

Metallography is crucial for a proper assessment of material properties. It mainly involves investigating the spatial distribution of grains and the occurrence and characteristics of inclusions or precipitates. This work presents a holistic few-shot artificial intelligence model for Quantitative Metallography, including Anomaly Detection, that automatically quantifies the degree of the anomaly of impurities in alloys. We suggest the following examination process: (1) deep semantic segmentation is performed on the inclusions (based on a suitable metallographic dataset of alloys and corresponding tags of inclusions), producing inclusions masks that are saved into a separated dataset. (2) Deep image inpainting is performed to fill the removed inclusions parts, resulting in 'clean' metallographic images, which contain the background of grains. (3) Grains' boundaries are marked using deep semantic segmentation (based on another metallographic dataset of alloys), producing boundaries that are ready for further inspection on the distribution of grains' size. (4) Deep anomaly detection and pattern recognition is performed on the inclusions masks to determine spatial, shape, and area anomaly detection of the inclusions. Finally, the end-to-end model recommends an expert on areas of interest for further examination. The physical result can re-tune the model according to the specific material at hand. Although the techniques presented here were developed for metallography analysis, most of them can be generalized to a broader set of microscopy problems that require automation. All source-codes as well as the datasets that were created for this work, are publicly available at https://github.com/Scientific-Computing-Lab-NRCN/MLography.

Material science and quantitative metallography. Material science is the study of material properties, which is based on, for example, understanding how it is influenced by its chemical composition, microstructure, and manufacturing process[1]. Metallography is the study of the microstructure of metallic alloys, or more generally of any material, in length scales usually ranging from nanometers to millimeters. Investigating the microstructure of a material allows one to discover important properties of that material, such as mechanical properties[2–4], corrosion behavior[5–7], electrical properties[8–10], etc. Microstructure study is therefore considered to be among the most beneficial and effective fields for understanding materials' properties.

Different techniques[11] are used to reveal various microstructural features of metals. Most investigations are carried out with incident light microscopy dedicated to metallography (metallographic microscope), which can operate in different modes such as bright field, dark field, polarized light, etc. Another common technique for metallographic investigation is Scanning Electrons Microscopy (SEM). This method is based on electron beam emission from an electron gun which interacts with the sample, then monitoring the different signals resulting from this interaction, mostly backscattered electrons, secondary electrons, and characteristic X-ray.

Metallography is usually performed on a flat sample sizing from a few millimeters to a few centimeters, and its' preparation includes mechanical grinding on silicon-carbide polishing paper and final polishing on clothes soaked with diamond paste. Sometimes this preparation is not sufficient for revealing the features of interest, and some extra preparation is required (such as chemical or electro-chemical etching or controlled oxidation).

Many correlations between microstructure and macroscopic properties have been explained using metallography. A notable example of such correlation is the Hall-Petch strengthening mechanism[12–14] in which there is a negative correlation between the yield strength of a material and the square root of its' average grain size. Another

[1]Scientific Computing Center, Nuclear Research Center-Negev, Be'er-Sheva, Israel. [2]Department of Physics, Nuclear Research Center-Negev, Be'er-Sheva, Israel. [3]Department of Materials, Nuclear Research Center-Negev, Be'er-Sheva, Israel. [4]Department of Computer Science, Technion-Israel Institute of Technology, Haifa, Israel. ✉email: galoren@cs.technion.ac.il

example is the correlation between the nature and concentration of inclusions in some material to its' mechanical endurance under different loads, such as under quasi–static loading[15,16] or fatigue (cyclic loading)[15,17]. The ability of a material to hold quasi–static loading is relevant to all aspects of life: from the capability of buildings and bridges to endure loads to the strength of a kitchen knife. Fatigue is an important issue when dealing with moving parts such as shafts and motors[18,19], and it has crucial safety aspects, for example, in the case of airplane wings and landing gear[20,21]. Another important macroscopic property is anisotropy. In most cases, isotropic properties (uniform behavior in all directions) are desired, but in some cases, anisotropy (material's tendency to react differently to stresses applied in different directions) is preferred. An example of such a case is jet engine blades, in which the ability to hold longitudinal (radial) load is more important than in the transverse direction.

As the material properties are highly affected by its microstructure, and the demand for superior materials' properties constantly increases for edge usage, safety reasons, or economic constraints, the need for tide controlling of the microstructure is increased as well. Hence, metallography is used in almost all stages during the lifetime of a component: from the initial materials development to inspection, production, manufacturing process control, and even failure analysis if required. The principles of metallography help to ensure product reliability.

Features of interest mainly and re-daily studied in metallography are the grains' spatial distribution and the occurrence and characteristics of inclusions or precipitates.

Grains are areas in the sample where the atoms are arranged in a specific crystallographic orientation. Grains are originated from phase transition, e.g., solidification, in which different areas in the material start to transform in different orientations and are growing until filling the entire volume of the material, or from recrystallization, where new grains of the same phase are growing on the expense of old ones. There are various reasons for the spatial distribution of grains. Different growing kinetics in different crystallographic orientations during phase transition and mechanical and thermo-mechanical shaping are among those. The interface between two adjacent grains is called a grain boundary (GB), and often it is the GB's that are seen in the metallographic image.

Inclusions and precipitates are small material areas that differ from their surrounding matrix either by composition, crystallographic structure, or both. Inclusions are originated from outside the sample, e.g., oxide particles that were added to the melt before solidification, while precipitates are formed from the sample itself, e.g., carbides formation in steels. Henceforth, we will define both of them as inclusions or impurities. Inclusions hold essential information that will be considered in this paper, including the nature of each inclusion (composition, crystallographic structure, size, and shape) as well as more general information such as the surface concentration of inclusions (on a cross-section) and their distribution.

During the development of new material, metallographic characterization is done comprehensively and deeply, while during routine manufacturing, it is usually done only once in a while to ensure the manufacturing stability[22]. In both cases, but more often during routine manufacturing, the metallographic images are analyzed by their similarity to previous samples[23]. This similarity is determined by Quantitative Metallography (QM) parameters such as grain size and the surface concentration of inclusions and also by 'softer' parameters such as shape and distribution of the inclusions, which are usually based on an 'expert opinion'[24]. Based on computational algorithms, such an approach is less preferred than a more objective and quantitative one. Moreover, the expert is not always a qualitative source, as the ability to detect anomalies and analyze the material state quantitatively is a very complex task, and as such, it inherently requires and should heavily rely on computations.

However, for using these algorithms, it is first necessary to identify each relevant feature in the metallographic picture, specifically the inclusions and the GB's. In many cases, this is not a trivial task. For example, in Fig. 4a a metallographic image of U-0.1%Cr sample is shown (this alloy is used as a nuclear fuel, and as such, the importance of repeatable manufacturing is substantial). It can be seen that the contrast between the inclusions and the matrix is not uniform and that the GB's are not easily noticed. Because of this complexity, classic image processing tools did not yield satisfactory results, and hence the identification of these features is made manually by an expert as an input for the computational analysis. This procedure is very tedious and time-consuming and does not allow the analysis of a high number of pictures for improved statistics.

Previous work.   As metallographic imaging heavily relies on semantic understanding of complex combinations of colors, boundaries, structures, and their corresponding distributions, many image processing and computer vision techniques were introduced over the last decade to extract the needed characteristics automatically. The need for such automation is well motivated from the material science perspective[25–31], as well as from the perspective of microscopic imaging in general[32,33]. In relatively-simple cases, classic algorithms for image pre-processing, pixel classification, region extraction, and edge detection, occasionally combined with data mining methods, proved to yield moderate to good results[34–40]. However, as the image complexity and resolution increases, and the image domain varies, the error rate in the needed tasks using those methods increases as well, thus demanding human expertise in tuning said algorithms[34]. For example, in cases where segmentation depends on the color histogram or some hyperparameters of other edge detection algorithms, the expert is required to tune the algorithm until reaching the desired result[34].

Nevertheless, as many alloys—even of the same materials and under similar conditions—exhibit extreme variance in colors, boundaries, structures, and corresponding distributions. Therefore the tasks of classifying the different objects (grains and inclusions) and segmenting those objects correctly (where and how to define the borders) almost completely depends on human expertise, which often cannot precisely determine the accurate classification nor the border edges[41–43]. Thus, in recent years the introduction of computer vision-based on deep learning (CVDL) techniques[44–48] to the QM field is growing, as those techniques proved to minimize the error rate of previously exclusive human tasks, and sometimes to even introduce new capabilities. For example, in the task of image inpainting (inpainting of inclusions)—by using generative models such as Generative Adversarial Networks (GANs)[47].

The need for such segmentation and knowledge of one sample's properties is crucial, and many features can be extracted that way to understand the state of the material. However, reliable statistics, anomaly detection, and patterns recognition cannot be gained sufficiently in this form, and there is a need to quantify the material's features and metric their behavior over a considerable amount of samples[49], which is far greater than human comprehensibility in the age of Big Data[50]. For example, material scientists often rely on some quantifications in order to provide a quantitative explanation of how anomalous each impurity/whole sample is, such as the distribution of grains' size or the total sum of impurities' area, and comparing those to previously gained results[51,52]. While those are essential features, they hardly represent the material status, which should be quantified for each object in terms of spatial, shape, and area distribution in the lone sample and the entire history combined[53,54].

This methodical and statistical observation of the materials' properties is also crucial in the perspective of Quality Control (QC) and Quality Assurance (QA) in order to meet repeatable manufacturing standards[55]. The goal of quality and reliability assurance in materials' research and development is the continuous improvement of performance, efficiency, and ease-of-use of the formed material[55]. The acquisition and analysis of quantitative observations made during the process of production allow it to be continuously monitored and maintained to meet the desired specifications, or ultimately rapidly diagnosed and remedied if problems arise and the results deviate from the expected ones[55]. As metallography plays a key role in understanding materials and alloys' behavior under production, the need to automatically and reliably diagnose metallography while comparing it to the previous diagnosis and quantify the changes is necessary. Those requirements can only be met using novel data mining and machine learning approaches—specifically in the field of anomaly detection and pattern recognition—which can automatically gain the needed insights in a reliable fashion[56].

Contribution. The benefits of our proposed methodology versus the above-mentioned existing works are rooted, first and foremost, in the ability to bind in a pipeline fashion all of the building blocks of automatic QM while entirely relying on artificial intelligence in general and CVDL in particular. Specifically, we propose a segmentation pipeline in Fig. 1 and an anomaly detection pipeline in Fig. 2. The segmentation pipeline consists of (1) deep semantic segmentation; (2) deep image inpainting, resulting in 'clean' metallographic images; and (3) grains' boundaries deep semantic segmentation. The impurities anomaly detection pipeline consists of (4) spatial anomaly detection; (5) shape anomaly detection; and (6) area anomaly detection. To the best of our knowledge, this is the first introduction of such a pipeline, including a vast state-of-the-art human expertise labeled QM dataset based on actual metallurgy (see "Dataset creation"). This pipeline was established while reinventing each process step technology to adjust the problem and even devise new computer science methods. Among those, we include the weighted variant of Kth-*Nearest-Neighbor*, a novel training methodology for Auto-Encoders that favors normal examples while discriminating abnormal ones, and the novel *Market-Clustering* anomaly detection algorithm, that are all based on the physical distribution and properties on the material. Moreover, our usage of immensely complicated images, with many different colors, shapes, and textures, while using only a tiny portion of this data for training (about 1%), strongly suggests that those findings can be replicated easily for any other material with minimal effort. An additional novel concept is our end-to-end QM model's ability to constantly calibrate itself with respect to new experiments and not solely rely on previous knowledge (Fig. 3). This way, we enhance and adjust the pipeline regarding the actual state in the lab or the factory.

Dataset creation. While there are many datasets of electron microscopy, almost none of them is suited for machine learning purposes[57,58], and virtually none of those is suited to the goal of a unified automated approach to the entire metallographic investigation process (Fig. 3), including labeling of impurities and GB's over different and complex backgrounds. This state is not a surprise, as the common hypothesis among material science practitioners is that this task is inherently tricky and prone to ambiguation[59]: the data itself is not natural, includes many textures, and primarily represents a 2D cross-section of a 3D reality (see Appendix A for more examples). Furthermore, as only experts can label this data, the labeled data is naturally scarce, and there is no real option to reach labeled big data. As a result, our goal in this work was to establish an end-to-end model on *as minimal as possible labeled images*. Iteratively, we added batches of labeled images until reaching the desired performance, all of this while designing the model as a few-shot model. In fact, only 1% of the data that eventually been evaluated was labeled. Even in comparison to U-net initial dataset[60] we used half as much labeled data, while our complexity is much greater. We managed to do so by the usage of sliding window over the scans, redesigning U-net (see "Impurities' segmentation"), and performing physics-oriented post-process (see "Grains' boundary segmentation").

The outcome of this iterative process resulted in the open-sourced *MLography* dataset, which includes extremely detailed metallographic images, with all of the features for QM labeled by experts. Specifically, we used the U-0.1wt%Cr alloy, which is used as a nuclear fuel. In this alloy, the most abundance inclusion is *Uranium Carbide*, which appears on the 2D metallographic cross-section as dots, spots, or long rods, depending on the impurity concentration in the alloy and the thermal profile during casting and cooling to room temperature[61]. For this work, metallographic images (approximately 1.2 mm × 0.8mm) were used (see example in Fig. 4a). An expert tagged each inclusion and its boundaries on it (see example in Fig. 4b,c). As these kinds of datasets are rarely public, except for a few exceptions[62], we have created a novel dataset of manual tags of impurities from 243 metallographic scans for anomaly detection, and we make it publicly available at[63]. We present our anomaly detection results on a sample image of tagged impurities from a uranium-chromium alloy scan from the dataset in Fig. 16a–f. Additionally, we have created two datasets for metallographic semantic segmentation, consisting of cropped squares (128 × 128 pixels) of metallographic scans with corresponding manual tags of impurities as ground truth (261 images in[64]), as well as a dataset of cropped squares with manual tags of grains' boundary as
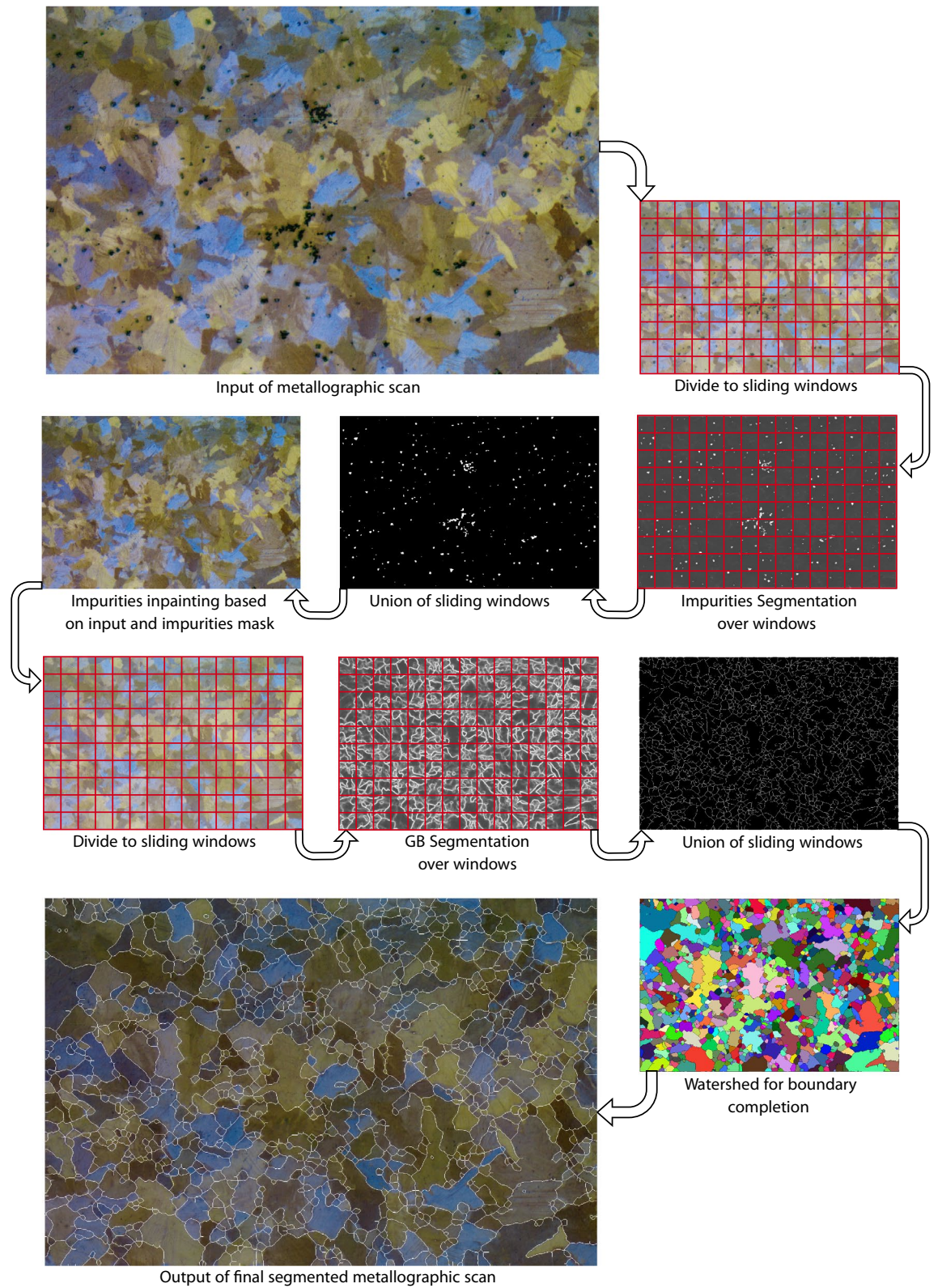
Figure 1. Proposed segmentation pipeline. This pipeline serves, in part, as the data pre-processing to the anomaly detection pipeline.
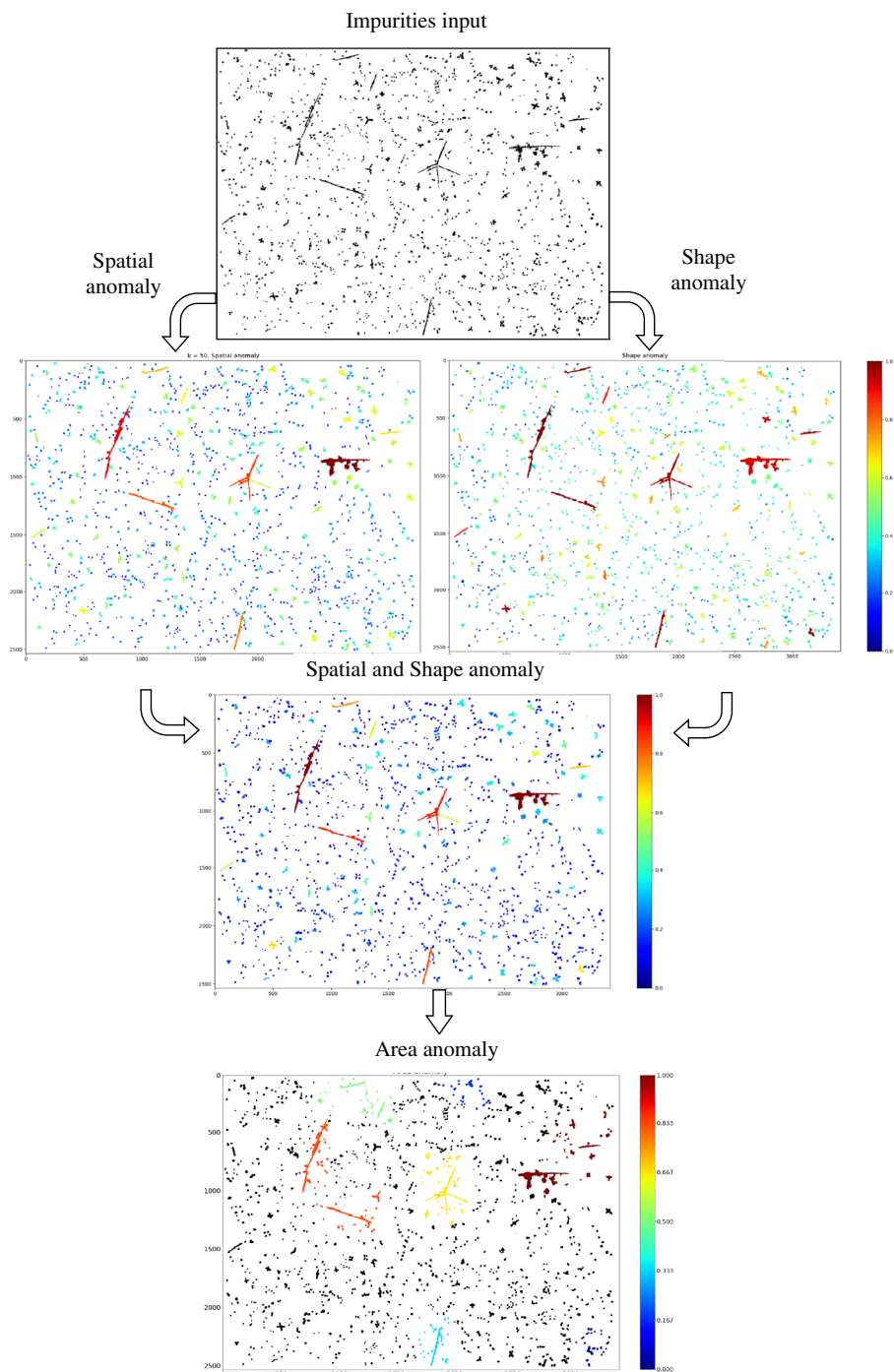
Figure 2. Proposed anomaly detection pipeline.

ground truth (320 images in[65]). We also provide unlabeled 32 big metallographic images in[66] and present the output of the segmentation pipeline in[67].

Paper organization. The rest of the paper is organized in accordance to the methodology pipeline: Section "Segmentation pipeline" presents the segmentation pipeline including impurities' segmentation ("Impurities' segmentation"); generative inpainting over the segmented impurities, creating 'clean' metallographic images ("Impurities' inpainting"); and grains' boundary segmentation on the 'clean' images ("Grains' boundary segmentation"). In "Segmentation pipeline evaluation", we evaluate this entire pipeline as a whole. Next, in "Anomaly detection measures", we introduce new anomaly detection measures for metallography. Finally, in "Anomaly detection evaluation—tuning via feedback", we suggest a new *physical* evaluation to the presented methodology using both computerized and mechanical measures.
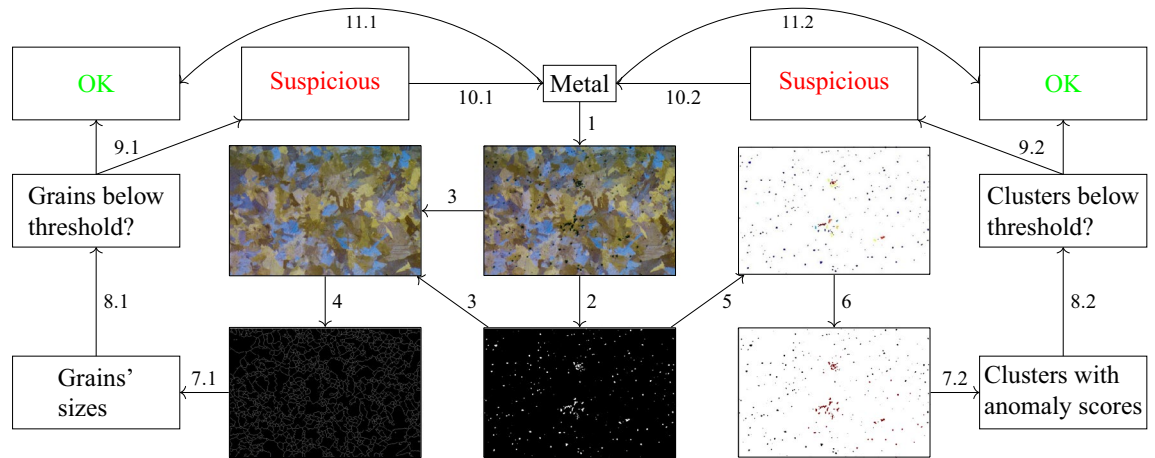
Figure 3. The workflow of our proposed end-to-end model for QM on a given input metal. First, in 1, metallographic imaging is performed. Then, in 2, semantic segmentation for impurities is carried out. Using the metallographic image and the mask of segmented impurities, the impurities are inpainted in 3, creating a 'clean' metallographic image. Next, grains' boundaries are semantically segmented over the metallographic image without the impurities in 4. In 5, spatial and shape anomaly detection measures are calculated from the segmented impurities, and in 6, the anomalies are clustered based on area anomaly. In 7.1, the grains' sizes are calculated, and in 7.2, each cluster of anomalous impurities receives a numeric value representing its anomaly score. In 8.1 and 8.2, statistics based on these values are calculated and compared against pre-set or learned corresponding thresholds. These thresholds determine in 9.1, 9.2 whether the input metal is OK (under the threshold) or suspicious (over the threshold). If one of the values exceeds the threshold, a suitable physical test is performed (10.1, 10.2) on the input metal in the advised area from the model. If the physical test establishes no fault in the material, the sample is set as OK and the corresponding threshold is tuned respectively in 11.1 and 11.2.
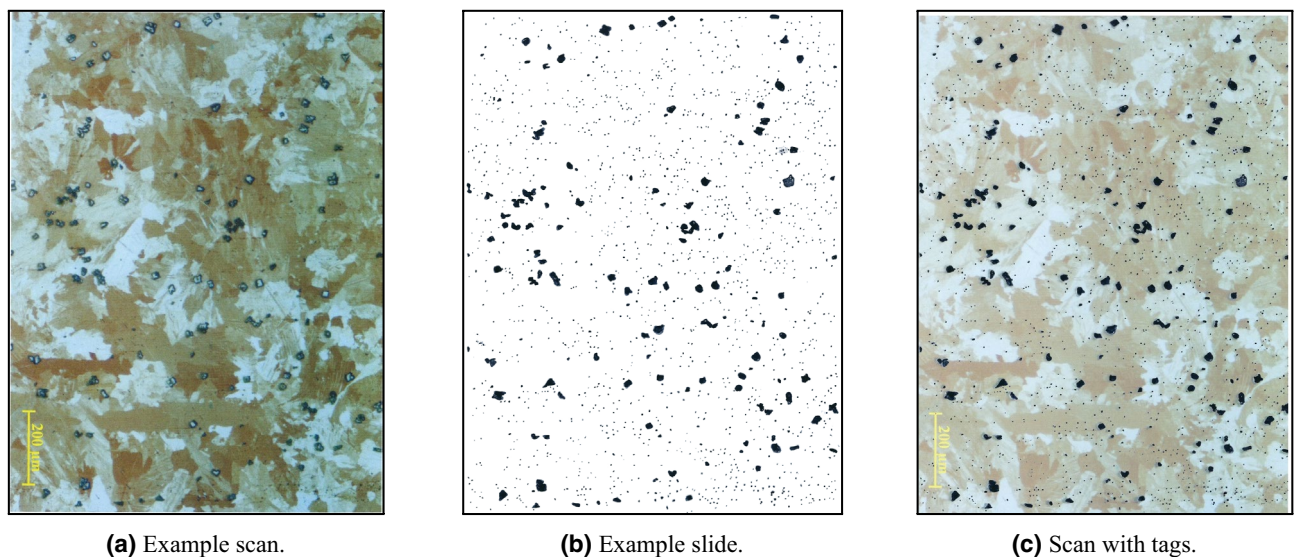

(a) Example scan. **(b)** Example slide. **(c)** Scan with tags.

Figure 4. Example of metallographic sample with impurities tags.

Segmentation pipeline

Impurities' segmentation. As we stated, the above anomaly detection measures are only applicable to image samples representing a slide containing the tagged impurities. The two main approaches for achieving these images are: providing manual and pricey expert-made tags; generating machine-automotive tags from metallographic scans. The latter might be a much more efficient alternative as it allows the experts to save precious time for other work while completing the task of tagging the impurities in a much faster way. However, it is risky to produce non-accurate impurities tags, as this is a delicate task that involves determining whether each pixel is an impurity or a part of the background.

The above task is called Semantic Segmentation, and it is a well-studied topic in Computer Vision. Due to the emergence of Deep Convolutional Neural Networks (CNN) in recent years, lots of CNN models were proposed[68],
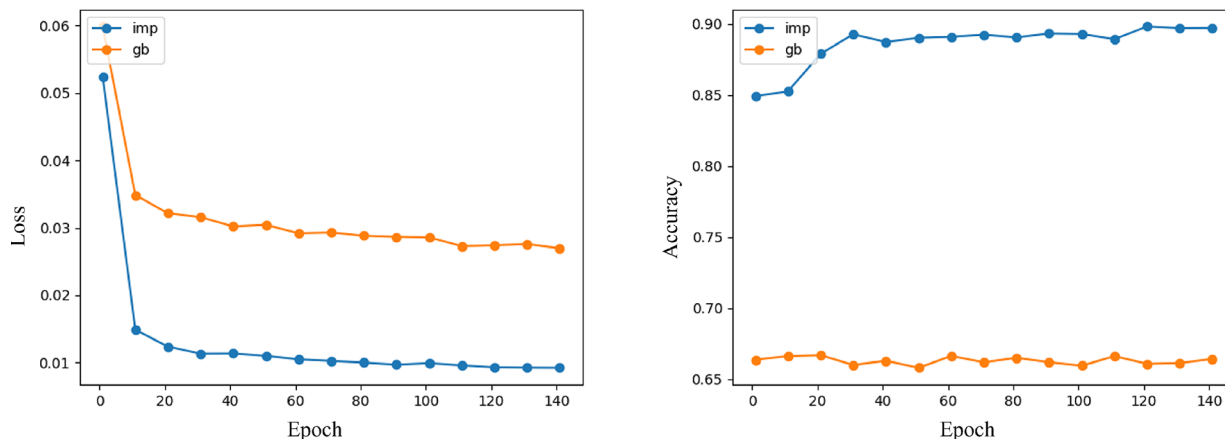
(a) Training loss. Impurities Segmentation in blue, GB segmentation in orange.

(b) Training accuracy. Impurities Segmentation in blue, GB segmentation in orange.

Figure 5. Training loss and accuracy of the segmentation tasks.

achieving state-of-the-art performance for segmentation. U-Net[60] is an example of such a model that is specifically tailored for the segmentation of microscopic biomedical images that resemble our images' characteristics. U-Net's architecture consists of an Encoder that extracts and progresses the underlying abstract information from the images to the Decoder. The latter utilizes the *global* information (low-resolution details such as structures) by up-sampling the latent information from the encoder and the *local* information (high-resolution fine details such as textures) directly from the corresponding layers from the encoder. This 'skip' mechanism allows learning deep semantic information while preserving high-resolution shallow information that might get lost due to down-sampling and up-sampling that CNNs require for the training process.

U-Net highly inspires the architecture of our model, but it has some modifications. First, we use VGG16[69], that was pre-trained on ImageNet as the encoder (similarly to[70]). It receives colored input images of size 128×128, and it passes the latent information to a symmetric decoder network, making the entire model look like a U-shaped network with a total number of 27 convolutional layers. It was optimized using Adam and with the focal loss function[71] with $\alpha = 0.2$ in order to penalize cases in which the model misclassifies a pixel as a background. We trained the model on small images (128×128) that were labeled using[72,73], for 150 epochs—300 steps in each. Training loss and accuracy trends are presented in blue in Fig. 5a,b.

In order to be able to segment images bigger than the small images that the network was trained on, we propose the following methodology that extends the compatibility of the model to images of any dimensions and improves the performance of the model significantly. We suggest using a sliding window with the same size as the inputs of the network, with *high-overlap* in order to minimize noise and to increase the certainty of the segmentation. This way, each window is segmented using the network, and each pixel in the output of the full segmented image is determined via an average of all pixels in the corresponding segmented windows that contain him. As we can see in Fig. 6e, the network can not infer the true nature of the objects in the borders of the images since it does not have the 'full picture.' Therefore, we see that the networks have small confidence that impurities may reside in the borders. Using the high-overlapping sliding-window methodology helps with this issue: the right edge of an impurity can be seen in the left border of Fig. 6e. Then, since several windows will contain this impurity in its entirety (e.g., Fig. 6d), they should agree on segmenting it, and thus after averaging the pixels of all segmented windows, this impurity should be classified as an impurity. Following a similar logic, noise reduction might be achieved since if there is no real impurity in the left border of the window, the same windows should agree via *majority vote* on not segmenting any of the corresponding pixels. The results of the model on Several examples are presented in Fig. 6a–f. For fast applications, we suggest tuning the stride of the sliding windows, reducing the overlap, and lowering the running time of the application. However, this running-time reduction might come with performance degradation, since *majority vote* is crucial, especially if the network was trained upon such a small dataset as in our case. Therefore, we suggest the user address this trade-off and set the stride parameter according to the application.

In order to increase the model segmentation resolution, we suggest focusing the model on finer details by zooming-in the picture prior to the methodology described above. The final segmentation result using windows of size 128×128, 8 pixels offset between every two consecutive windows, and a zoom-in factor of 3 on some input scan (Fig. 7a) can be seen in Fig. 7b.

Evaluation. In order to evaluate the proposed U-Net architecture, we trained another network with the same architecture on 95% of the dataset (247 squares), leaving a test set of 14 squares. The Receiver Characteristic Operator—Area Under the Curve (ROC-AUC) score is 0.96, implying that the network classifies pixels very well. The mean Intersection over Union (IoU) score of the impurities' bounding boxes is 0.73. IoU severely penalizes
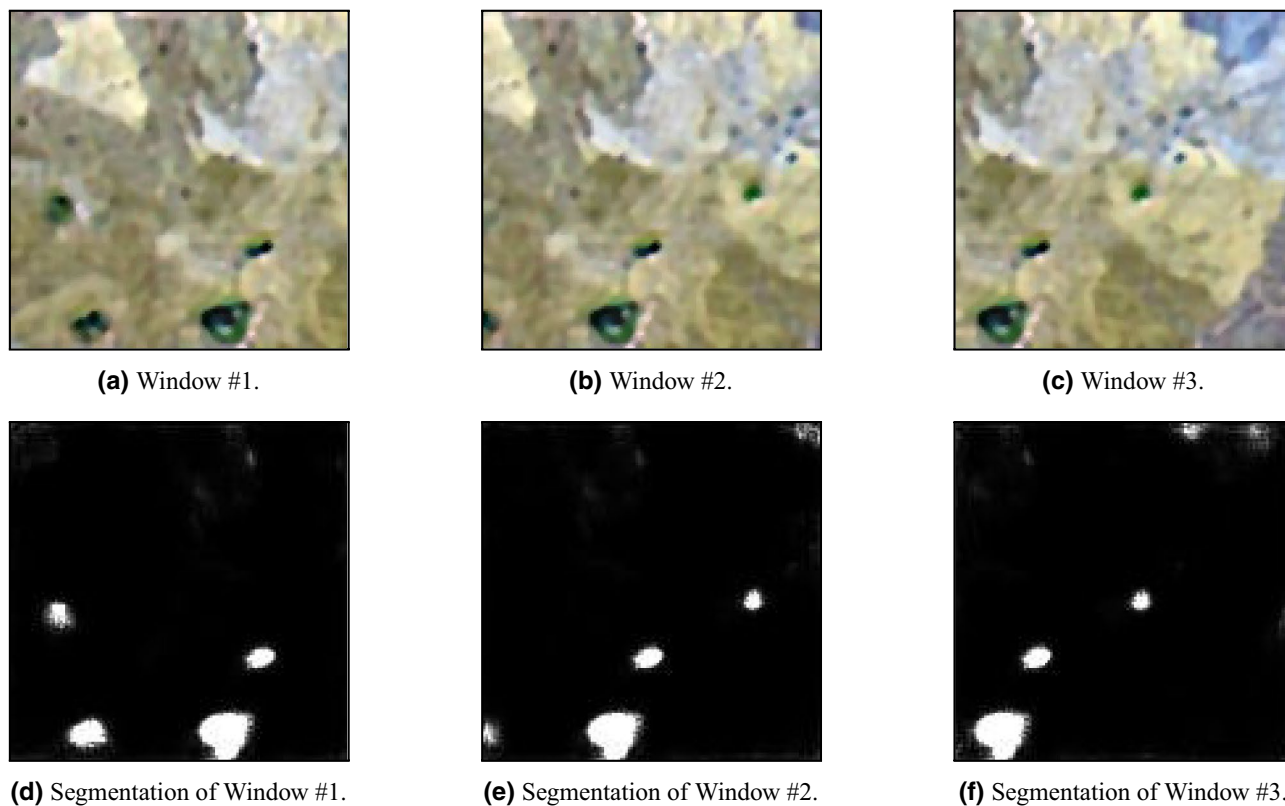
(a) Window #1.   **(b)** Window #2.   **(c)** Window #3.

(d) Segmentation of Window #1.   **(e)** Segmentation of Window #2.   **(f)** Segmentation of Window #3.

Figure 6. Squares of impurities and their segmentation.
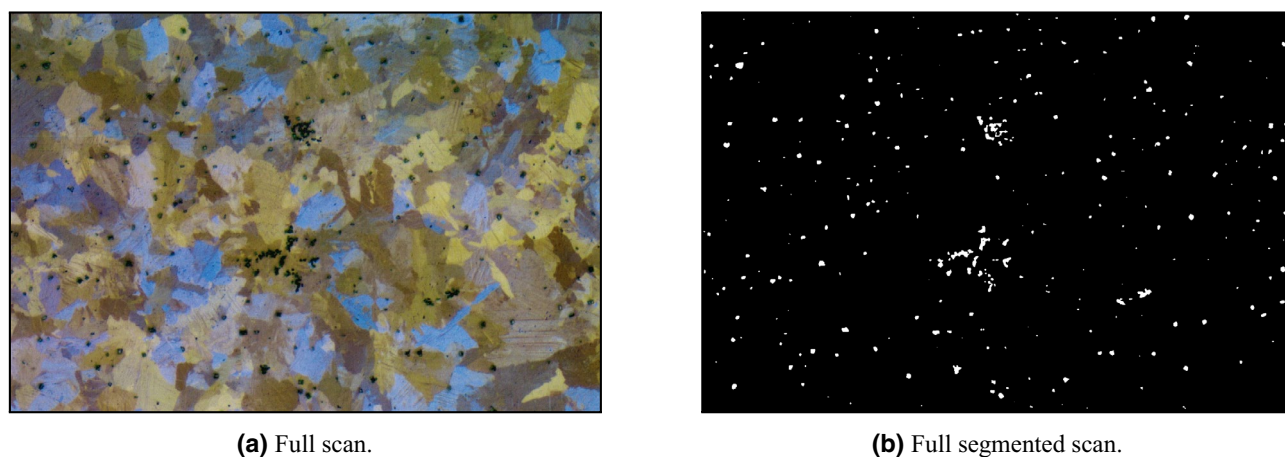

(a) Full scan.   **(b)** Full segmented scan.

Figure 7. Impurities segmentation.

misclassifications of a few pixels since the final score is normalized over the union of segmented pixels, and the number of pixels that correspond to impurities is usually tiny compared to the size of the image. Moreover, in most cases of QM, it is much more crucial not to neglect big impurities than to suffer from misclassifications of a few pixels. We present an alternative measure that counts the number of intersections of the bounding boxes of the impurities over the number of impurities (maximum between the number of impurities in the prediction and the ground truth). This measure can be interpreted as Object Intersection over Union (OIoU), suitable for object localization. The mean OIoU on our test set is 0.85. The mean percentage of impurities from the squares in the predicted images is 8, while the ground truth is: 7.

Impurities' inpainting. Once the exact placements of the impurities from a given metallographic scan are achieved, one can try to fill them with synthetic 'normal' parts such that the new image will resemble a realistic full metallographic scan without any impurities. Several works purpose solution for this task[74,75]. We tested both models and found the pre-trained generative inpainting model[75] to be very effective on the impurities masks
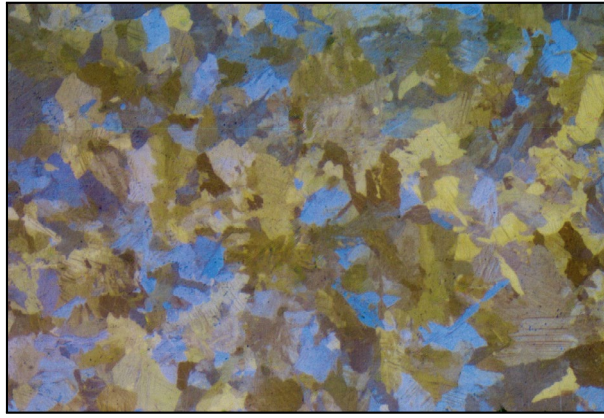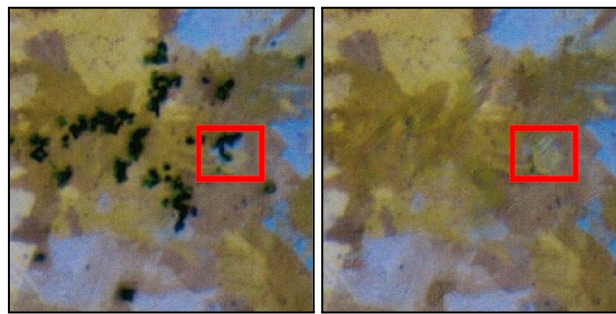
Figure 8. Inpainted impurities.

(a) With impurities.   **(b)** Without impurities.

Figure 9. Zoomed portion with and without impurities. The blue grain in the red square disappeared in the inpainted image.

(after applying a dilation filter for border expansion). The result on the full metallographic image is presented in Fig. 8, while zoomed-in images representing a small portion of the image with and without impurities are presented in Fig. 9. Although the result seems to be satisfactory, the network struggles to complete small grains (areas of the same color) that are shadowed almost entirely by impurities. For example, a small blue grain in the center of the images in Fig. 9 has disappeared and 'collided' into the yellow grain below it. This issue might be fixed by optimizing the network with a tailor-made annotated dataset. However, as we will later explain, small modifications in the borders of the grains, or even collisions of small grains into other grains, usually are not meaningful since metallographic examination of GBs often involves statistical analysis that neglects small differences.

Evaluation. Since the generative inpainting model was not trained on our metallographic data, we could test its performance on it entirely. We applied the model on 32 couples of metallographic images, I_i, and their corresponding impurities masks, M_i. For each couple of I_i, M_i, the model generated clean metallographic images: C_i. Then, random crops of 512×512 pixels were taken from C_j and M_i, denote these crops as C'_i and M'_i respectively. The model returned a prediction P_i for each couple C'_i, M'_i. We report the mean Peak Signal-to-Noise Ratio (PSNR) value between all P_i and C'_i to be 34.68, while in[75] the reported PSNR value was 18.91. A possible explanation for the higher performance in our case is that the impurities mask spans a relatively much smaller region than in[75].

Grains' boundary segmentation. Another critical aspect of metallography is the study of the spatial distribution of grains. Similar to the task of anomaly detection for impurities, one must first detect the grains or their GBs as a preliminary stage. However, differently from the former case, impurities' segmentation deals with localization and segmentation of pixels belonging to the class of impurities. Impurities might be characterized by several discriminative properties, e.g., continuous untextured zones of pixels with a dark homogeneous color in relation to grains' heterogeneous (color-wise and texture-wise) background. On the other hand, GB segmentation deals with a much more complicated problem. Unlike learning discriminative features of impurities, the objective in GB segmentation requires learning and identifying differences between adjacent grains, specifically regarding their edges. Since grains come with a wide range of colors and textures, achieving a generalization is more challenging.
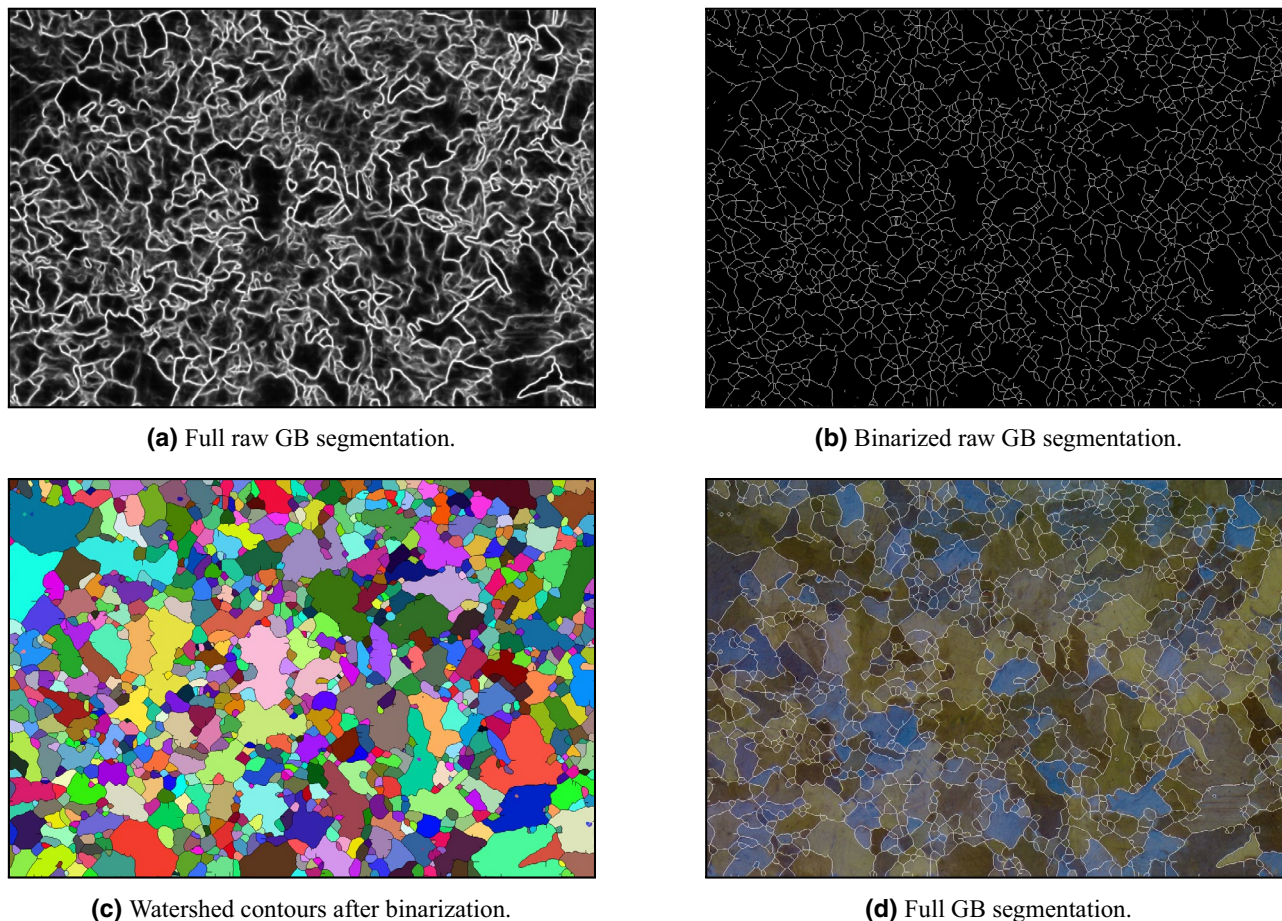
(a) Full raw GB segmentation.

(b) Binarized raw GB segmentation.

(c) Watershed contours after binarization.

(d) Full GB segmentation.

Figure 10. GB segmentation.

Edge detection is a well-formulated problem analogous to GB segmentation since, in both cases, the aim is to identify points in a digital image at which there are discontinuities or sharp changes in image intensity. However, grains might accommodate sharp intrinsic intensity changes that should not be considered as edges since they are defined as parts of the grain. Moreover, these intrinsic discontinuities can sometimes be even more intense than the discontinuities found on the GB. Therefore, naïve image processing approaches that search for intensity changes without a deep understanding of the image, such as Canny[76] were found to be not satisfactory for GB segmentation. On the contrary, W-net[77] is a self-supervised deep neural network that is trained to segment edges based on labels that represent sharp intensity changes, who found to be not satisfactory for GB segmentation for the same reasons as above. We also tested DexiNed[78] which is a state-of-the-art edge detection model that was pre-trained on a dataset of natural images with human labels named BIPED. Although DexiNed is optimized to understand deep features from images for edge detection, we found it not suitable for GB segmentation.

From the reasons above, we generated a dataset with manually annotated GBs and optimized a U-net model with the same architecture as in "Impurities' segmentation" against it. Training loss and accuracy trends are presented in orange in Fig. 5a,b. The raw output of the model on Fig. 8 as an input, is presented in Fig. 10a. However, since the model's output is a prediction on each pixel, there is often a variance on the predicted values on different boundaries and even on the same consecutive boundary 'line'. As a result, some predicted 'boundaries' appear as incomplete lines. Since these incomplete lines represent the uncertainty of the model, we interpret them as noise. We stress that this decision is based on the working methodology of material scientists. That is to say, if material scientists were to mark any suspicious boundary (and not only the 'certain' boundaries)—these incomplete lines would not have been neglected and treated as noise.

In order to suppress the noise, a few post-processing steps are required. A binarization, followed by Guo-Hall thinning[79] steps are applied (Fig. 10b). Then, Watershed algorithm[80] is used to segment grains with complete boundaries. This step allows us to eliminate incomplete lines and persist only 'certain' boundaries. The contours that were generated from the Watershed algorithm are presented in Fig. 10c, and the input image marked with the post-processed boundaries is presented in Fig. 10d.

We note that from the perspective of repeatable manufacturing and the need for reproducibility, uniformity in the metallographic examination process must be achieved. An agreed method of quantifying a metallographic sample is by inspecting the distribution of grains' size[41]. Therefore, the benefits of the proposed procedure are quantitative and qualitative: It shortens the time-consuming, demanding, and most of the time *subjective* task of

manually annotating GBs, but it also regulates and standardizes the process based on 'few' commonly accepted training examples of annotated GB's.

Evaluation.   Like the impurities segmentation task, we trained another network with the same architecture on 95% of the dataset (304 squares), leaving a test set of 16 squares. The ROC-AUC score is 0.84, and the IoU is 0.68, showcasing the complexity of the GB segmentation task compared to the impurities segmentation.

Segmentation pipeline evaluation

Different from standard evaluation procedures of classic machine learning tasks such as classification, segmentation, and inpainting—that were previously described with their corresponding evaluations metrics individually—there is also a need to evaluate the fusion of those models for the physical domain original purposes. Specifically, it would be of interest to evaluate the assembly of some or all of the presented algorithmic techniques as a QM process. That is, to test the decomposition of a given metallographic input, on which one or several algorithmic steps are applied (i.e., impurities and GB segmentation), followed by recomposition into some metallographic insight. For the segmentation task, we can examine the composition of the three models of "Impurities' segmentation"–"Grains' boundary segmentation" and their post-process procedure to match the guidelines of current standards in the field[22–24,41–43].

We evaluated both impurities and GB segmentation tasks with ROC-AUC statistic on a test scan of size 1328×896 (Fig. 11a). The outputs of the impurities and GB segmentation tasks are presented in Fig. 11d,f their corresponding ground truths are presented in Fig. 11c,e. As can be seen in Fig. 11d our model tend to ignore small impurities. However, we note that these impurities are less important than the bigger ones and are neglected in our anomaly detection measures regardless. If desired, we suggest to introduce smaller impurities in the training set of the model. The ROC-AUC value for impurities segmentation is 0.944, while the value for GB segmentation is 0.868. The results emphasize the complexity gap between the two tasks, as was described in "Grains' boundary segmentation". To further test the impurities segmentation model, we compared the percentage of white pixels in the ground truth image as well as in the segmented impurities mask. In addition, for the GB segmentation model, we compared the average grain size via average grain diameter calculation based on a standard method[43], using the Image Pro Plus 6.0[81] software's dedicated module reporting the average length of 6 diameters at 5 degrees intervals around the centroid of each object. We note that the ground truth image of impurities contained only 0.32% more white pixels than the segmented mask of impurities and that the difference between the average sizes is under 8%. These gaps are acceptable since different human experts tend to mark impurities and grains' boundaries in similar discrepancies. An essential advantage of an automated procedure as the one we propose is that it regulates and standardizes the segmentation process based on a few commonly accepted examples used for training. Additionally, since the same algorithm can be used for all segmentation tasks, fewer inter-inconsistencies between different samples are expected. This step is in contrast to manual impurities and grains' boundaries marking, which is usually very challenging to achieve by a single human expert. The final output of the model, including impurities (segmentation and) inpainting and boundaries segmentation, is presented in Fig. 11b.

Anomaly detection measures

Spatial anomaly measure.   Unsupervised Distance-based is one of the most common setups for anomaly detection[82]. In this approach, an object is considered as an *outlier* based on its spatial properties. Most common among those properties is how distant the object is from its neighborhood. A unified distance-based notion of anomaly presented in[83]: *an object O in dataset T is a DB(p, D) − outlier if at least fraction p of the objects in T are ≥ distance D from O*, where *DB* stands for Distance-Based. Although this notion is applicable for generalizing statistical anomaly detection in distributions such as Normal, Exponential, and Poisson distributions, it lacks few crucial properties: it is not able to produce scores of an anomaly; It requires the user to provide the distance *D*; And most importantly it does not treat objects with shapes of a positive area, as the impurities in our study. Another common distance-based anomaly detection approach, Kth-Nearest-Neighbour[84,85] henceforth, Kth-NN, defines outliers by their distance from their *k*th nearest neighbor and sorts them by that measure. Indeed, this approach allows one to order an object by a measure that indicates how that object is *distant* from its neighborhood. Kth-NN was compared to other 18 different unsupervised anomaly detection algorithms on ten datasets, and it was found to outperform all other algorithms with regard to the accuracy, determinism, and the ability to detect global anomaly[82]. However, in this study, we focus on anomaly detection for geometric objects with a positive area with a high emphasis on their size, i.e., the desired spatial measure should consider the areas of the impurities in order to score each impurity by how it is *distant* and *big* compared to its neighborhood. To that end, we present a novel approach for *spatial anomaly detection for positive area geometric objects*. Our spatial anomaly detection approach first defines a pseudo-semi-metric distance function between two geometric objects by the distance between their *Straight Bounding Rectangles*. We use rectangles since they are simplistic and computationally easy to calculate for each impurity, yet accurate enough—both in terms of Contour Approximation as well as edges distance calculation. We note that a trivial approach for implementing this distance function might be using the euclidean distance between any two points on the objects, e.g., the centers of the objects as presented in Fig. 12a. Nevertheless, in the case of almost-intersecting two big objects, this approach will yield a much higher distance than what is expected since their borders are much closer than their centers (e.g., the distance between i_1 and i_3 in Fig. 12a). A similar argument can be made on any fixed points residing on the objects. Our function is summarized with the following 4 representative cases in Fig. 12b. The distance between an object i_1 to another object, in the case of non-intersecting rectangles (i_2, i_3, i_4), is defined by the shortest Euclidean distance between the boundaries of the two enclosing rectangles (i.e., the distance between the closest two edges in the
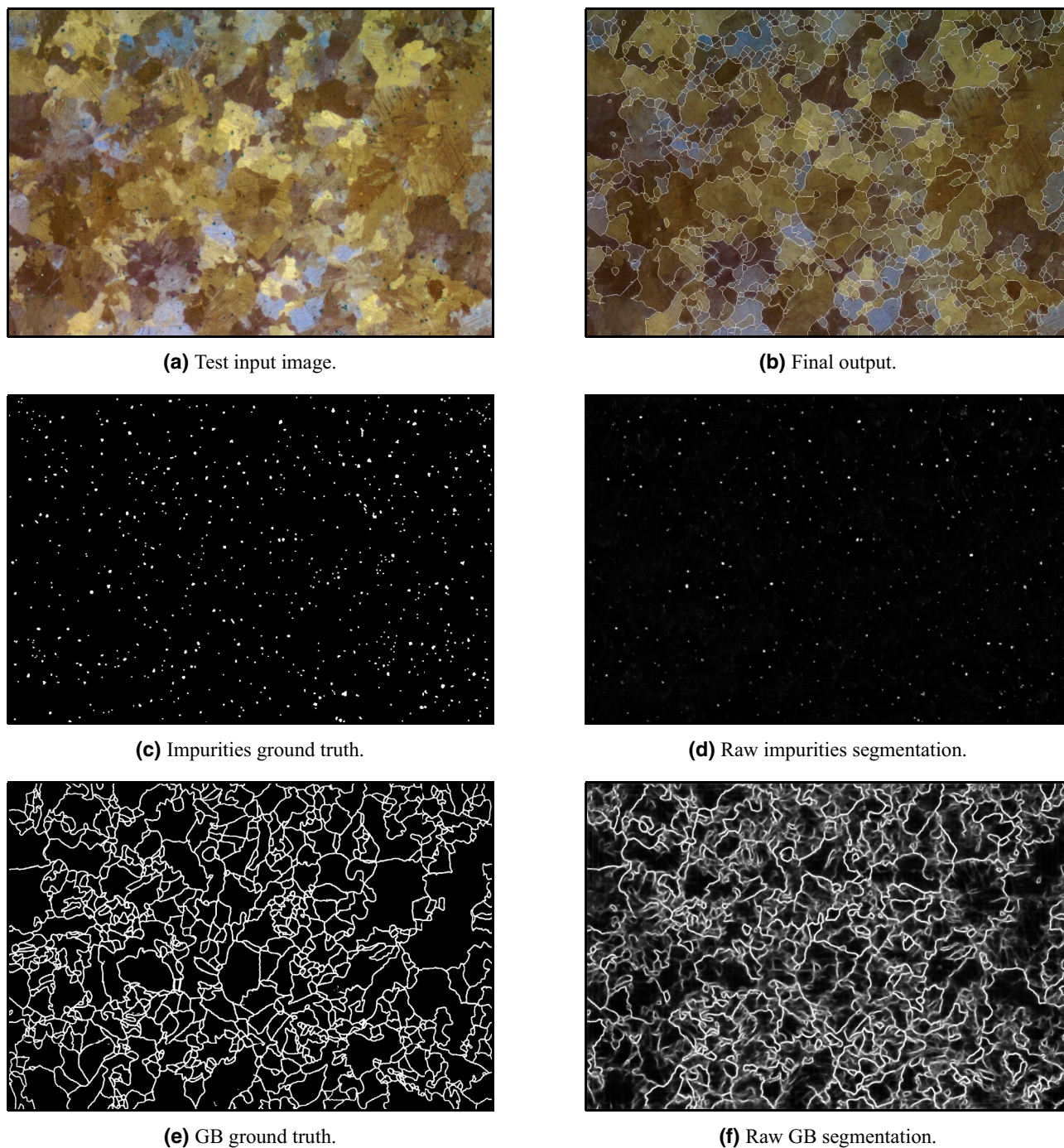
(a) Test input image.

(b) Final output.

(c) Impurities ground truth.

(d) Raw impurities segmentation.

(e) GB ground truth.

(f) Raw GB segmentation.

Figure 11. Segmentation pipeline applied on a test sample.

first two cases, and the distance between the closest corner vertices in the last case respectively). The distance between two intersecting enclosing rectangles (i_1, i_5) is simply defined as 0. It can be shown that this distance measure satisfies the symmetry axiom, and that for each two objects o_1, o_2 the distance is $d(o_1, o_2) \geq 0$ but the triangle inequality axiom is not met and not necessarily $d(o_1, o_2) = 0$ means that $o_1 = o_2$.

Next, we present a modified version of the classical Kth-NN algorithm[86]: *Weighted*-Kth-Nearest-Neighbor henceforth, WKth-NN, in which each object i refers to the distance between i to its neighborhood (defined above), along with the proportion between i's area to its neighbors. This modification allows having the spatial anomaly score to be calculated as a function of how the object i is *distant* from its neighbors, and also as how it is *big* compared to its neighbors. We now describe the algorithm, which is summarized in Algorithm 1. As in Kth-NN, the algorithm is parameterized by k – a constant that states how far is the neighbor from which we calculate the distance from. The procedure WEIGHTEDDIST calculates the weighted distance measure between the impurity i and the other impurity o. This measure combines the proportion between the areas of i and o and the distance

(a) Distance based on fixed point.　　　**(b)** Suggested distance function.
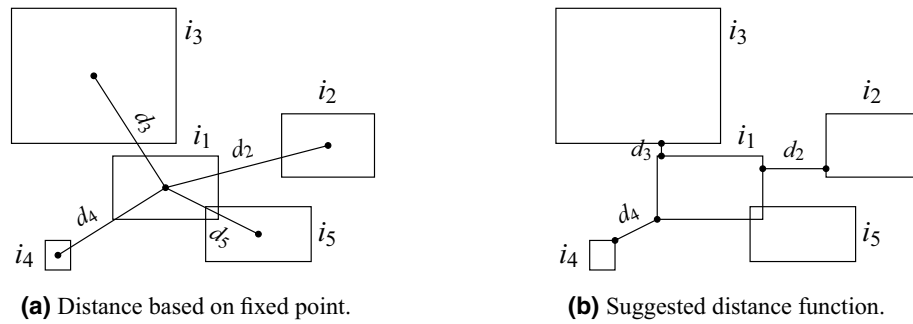
Figure 12. Possible distance functions.

between them (Fig. 12b). The main procedure, WKth NN, iterates over all objects (impurities in our case), $\mathscr{I}$, in line 2. For each object i, it calculates for all other objects o, the weighted distance in line 3. Then, it sorts the returned distances in line 4, and adds a factor of $Area(i)$ in line 5, in order to emphasize the significance of that object. Finally, when the iteration over all objects completes, we normalize and save the spatial scores of each object i in SS_i. The constants c_1, c_2 were set to 4, 2 respectively, but we encourage users to determine the values of the constants c_1, c_2, to suit best to their datasets. The output of the spatial anomaly detection algorithm on the input image with $k = 50$ is presented in Fig. 16a. The anomaly scores are normalized to $[0, 1]$, while the most anomalous impurities are with scores close to 1 and are colored in red, and the most non-anomalous impurities are with scores close to 0 and are colored in blue.

Algorithm 1 Weighted-KthNN

SS: list of size $|\mathscr{I}|$, SS as $SpatialScores$

Procedure WEIGHTEDDIST (i, o)

1　**return** $\left(\frac{\text{AREA}(i)}{\text{AREA}(o)} \right)^{c_1} * \text{IMPURITYDIST}(i, o)$;

Procedure WKthNN (k)

2　**for** $i \in \mathscr{I}$ **do**
3　　　$l = [\text{WEIGHTEDDIST}(i, o); \forall o \in \mathscr{I} \setminus \{i\}]$;
4　　　SORT(l) ;
5　　　$SS_i = \text{AREA}(i) * l[k]^{c_2}$;
6　$\forall ss \in SS, \frac{ss - min(SS)}{max(SS) - min(SS)}$;　　　　　　　　　　　　　　`// Min-Max norm`
7　**return** SS ;

Shape anomaly measure. Another crucial geometric object property is its shape, or how *close to some objective shape* is it, which in our case, how symmetric and how close the impurity is to a circle. Examining the output of the spatial anomaly detection algorithm may give the idea that spatial anomaly detection is sufficient for describing the degree of the anomaly in each object since it successfully marks objects that are clear to be anomalous with a high anomaly score. However, the spatial anomaly measure is not able to distinguish between an object that is not that big and distant compared to its neighborhood and does not have *anomalous* shape (e.g., an 'O' shape impurity), with an object of the same distance and size compared to its neighborhood, but with a much more anomalous shape (e.g., an 'X' shape impurity). Therefore, a consideration of the actual shape of each impurity is necessary to determine whether it is an outlier or not. A trivial measure for non-symmetric shape anomaly detection might be finding for each object i its smallest enclosing circle object, c (or some other basic geometric shape as in[87]) and setting i's shape anomaly score as:

$$\frac{Area(c) - Area(i)}{Area(c)}. \tag{1}$$

This measure indeed catches the most anomalous and non-anomalous objects based on their shape (i.e., impurities of a shape with area far smaller than their smallest enclosing circle's area, and impurities of a shape very close to a circle, respectively), but it fails to classify objects properly in the middle of the scale, as can be seen in Fig. 16b.

For example, impurities of the anomalous shape 'X' are marked only in the middle of the scale (e.g., the impurity within the black rectangle in Fig. 16b, which should get a higher shape anomaly score), together with not-that-anomalous ellipse-shaped impurities, as their shape's area is not that far from their smallest enclosing circle's area, although they should have appeared higher in the shape anomaly score scale. Indeed, Fig. 13 shows that there is a decent separation between the most anomalous impurities (scores ≥ 0.6) and the rest of the impurities (scores around 0.3), but the right tail of the distribution is quite long, which imposes noise to the model. Thus, from the non-linear nature of the problem at hand, we turned to train a Deep Convolutional Auto-Encoder
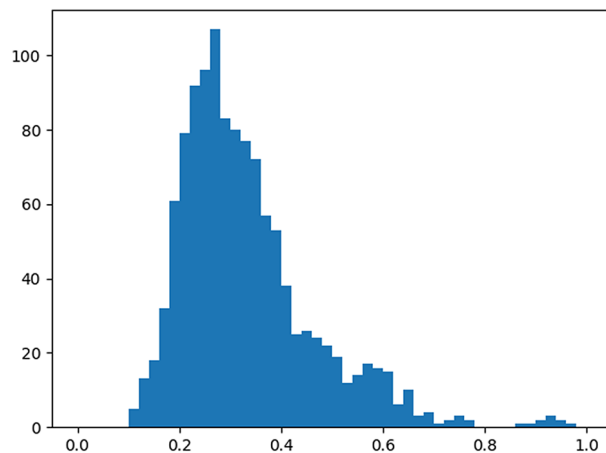
Figure 13. Histogram of circle difference scores in Fig. 16b.

Neural Network (henceforth, AE) for shape anomaly detection (also called Replicator Neural Networks)[88,89]. This method enforces the network to reconstruct images similar to the images from the training set and, hopefully, to fail to reconstruct images that are not similar to the images in the training set. As we already stated, the circle difference measure from Eq. (1) is a reasonable estimate for shape anomaly in the two ends of the scale—the most anomalous impurities and most non-anomalous impurities, which we will denote as normal impurities from now on. Thus, one can train an AE network in an *unsupervised* manner by providing the network, in its training phase, couples of all normal impurities as the training samples and copies of themselves as their labels. That is, fix a threshold for normal impurities and take all impurities with an anomaly score lower than that threshold.

We present in this work a novel approach to empower the separation capability of AE networks, in which, together with the normal couples of input and label images, the network is provided with couples of all the most anomalous images as input and blank images as labels. This method will urge the network to reconstruct the normal images successfully and to return a noisy-blank image upon an anomalous input, or in our case, anomalous impurity. This mechanism, in turn, will yield higher reconstruction loss for anomalous impurities, thus normalizing the reconstruction loss and using it as a shape anomaly measure will offer a higher separation between normal and abnormal impurities. We stress that one significant advantage of using neural networks is that it requires no assumption about the data. Therefore, one can employ the presented technique on any pre-defined 'normal' and 'abnormal' objects (in our case, difference from a circle). We set the threshold for normal impurities to 0.3 and anomalous impurities to 0.55, normalized and scaled all input images into the same size of 100×100 pixels. We note that albeit the size feature is not preserved in this measure, we still consider it in the spatial anomaly measure in "Spatial anomaly measure". Then we trained an AE network of the architecture presented in Fig. 14. We used different paddings for the convolutional layers in the decoder in order to reduce memory footprint and fit the model into 32 GB Tesla V100 GPU[90], using TensorFlow[91]. Loss and accuracy trends are presented in Fig. 15a,b. Yellow layers are Convolutional layers (5×5), Orange ones are Max-Pooling layers, Blue ones are Up-Sampling layers, Purple ones are Fully Connected layers, and the left and right-most images are example input and output images, respectively. In Table 1, we present the achieved reconstruction results on several use-cases, consisting of two normal-shaped impurities, $Norm_1$, $Norm_2$, and two anomalous-shaped impurities, $Anom_1$, $Anom_2$. Column *Image* holds the representative image of each impurity, and *Model 1 Recon'* holds the reconstructed image from the AE model trained on both normal and anomalous impurities. As we can see, there is a strong separation between the normal and anomalous impurities' reconstruction in the first model, as in the first two impurities, the reconstruction is a well-formed circle (with a varying intensity with respect to the degree of anomaly), and for the last ones, the reconstruction is a noisy-circle. For even sharper separation, we applied post-processing (threshold, erode-dilate) on the output of the AE, which is shown in *Model 1 Post-Recon'* column, and by that obtaining circles of different sizes for each of the normal impurities, and a blank image for the anomalous impurities. The column *Model 1 MSE* shows the Mean Squared Error (MSE) as the reconstruction loss between the input image and the reconstructed image after post-processing. As we can see, the first impurity is the most 'normal' impurity, and the last two impurities are much more anomalous. Conversely, the reconstruction results of the same input impurities on an AE trained only on normal set of impurities are presented in the columns *Model 2 Recon'*, *Model 2 Post-Recon'* and *Model 2 MSE*. As we can see, the intensity of the reconstructed circle scales negatively with the degree of anomaly, thus again yielding circles of different sizes in the post-processed reconstructions. Additionally, the MSE of the most anomalous impurity, $Anom_2$, is significantly higher than that of the most symmetric impurity, $Norm_1$, but the difference between the errors of $Norm_2$ and $Anom_1$ is mild. Thus the separation between the normal and anomalous impurities is flawed. Figure 16c,d present the output of both models. In each, the normalized reconstruction losses serve as the shape anomaly measure. The model that utilizes blank images as labels for anomalous impurities in the training phase significantly outperforms the second one since it has a more acute separation between normal-shaped and anomalous-shaped impurities, and it marks the anomalous 'X'-shaped impurities with a high anomaly score. We, therefore, use this model. The previously purposed spatial anomaly measure—combined by simple multiplication
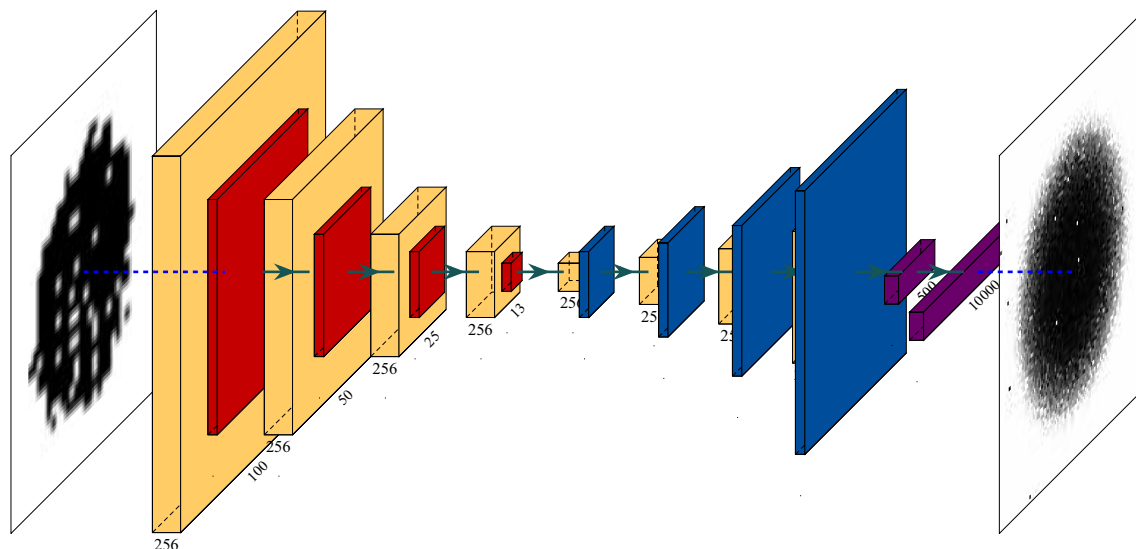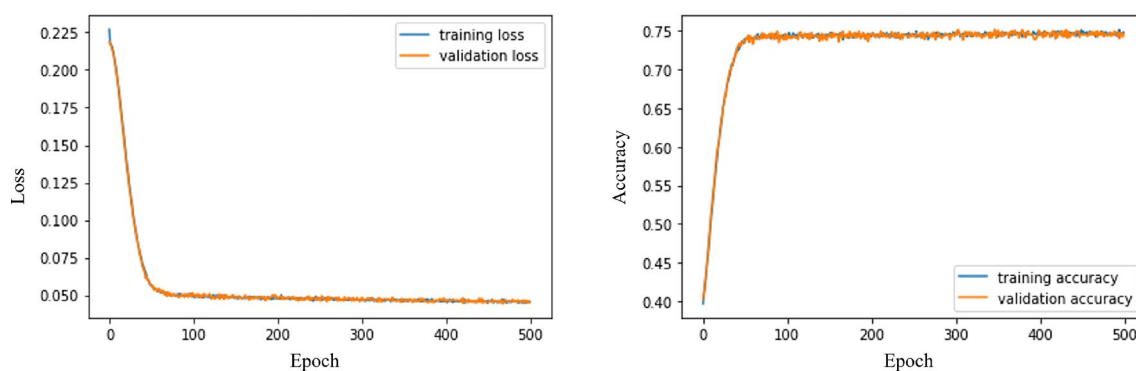
14

Figure 14. The AE architecture.

(a) Training and validation loss.

(b) Training and validation accuracy.

Figure 15. Loss and accuracy of the training and validation of the AE.

Name	Image	Model 1 Recon'	Model 1 Post Recon'	Model 1 MSE	Model 2 Recon'	Model 2 Post Recon'	Model 2 MSE
$Norm_1$				45,479			45,919
$Norm_2$				51,233			46,668
$Anom_1$				64,585			48,869
$Anom_2$				64,527			54,617

Table 1. Reconstruction of impurities in both AEs.

15

(a) Spatial Anomaly measure.

(b) Shape Anomaly with Eq. 1.

(c) AE trained with blank images as labels for anomalous impurities.

(d) AE trained with only normal images as labels.

(e) Shape&Spatial Anomaly.

(f) Area Anomaly measure.
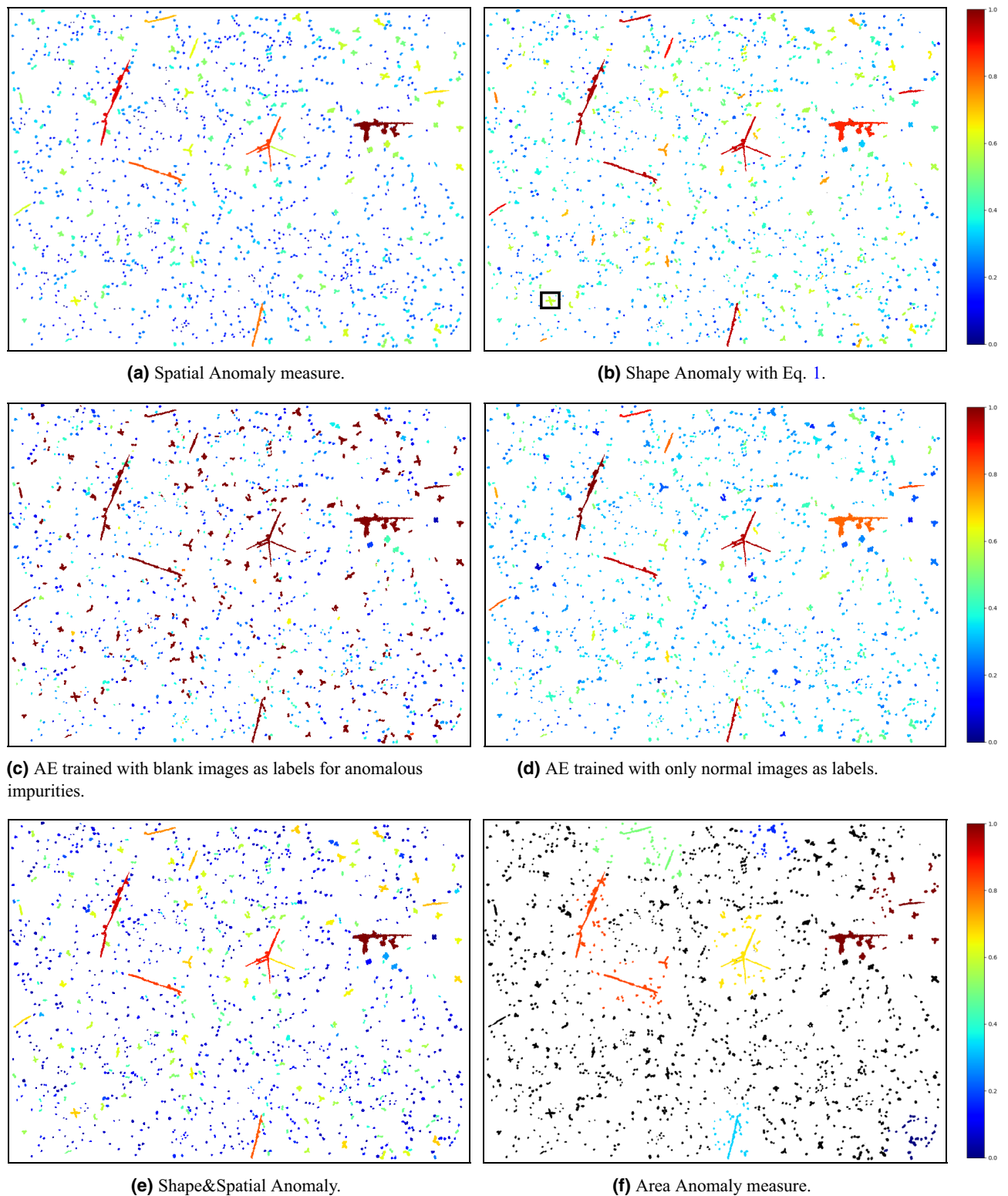
Figure 16. Anomaly detection measures.

and normalization with the shape anomaly measure—is presented in Fig. 16e. This measure significantly reduces the noise we had in the spatial anomaly measure while emphasizing the degree of the anomaly of anomalous impurities based on their shape and compared to their neighborhood.

Area anomaly measure. As previously explained, an essential application for anomaly detection in materials sciences is detecting defects. These defects usually span an anomalous *area* of several objects, rather than

just a single anomalous object[92]. For this reason, we present a novel clustering algorithm, which we call *Market-Clustering*, that divides impurities into anomalous areas, based on the anomaly scores of the impurities from the previous anomaly measures. The algorithm's name is inspired by the 'purchasing power' of each area/cluster and the economic decisions it should take to grow and merge with other significant clusters. In fact, each cluster's size, reach, and anomaly score is determined based on the anomaly score of the objects from the previous measures. The returned clusters are then ranked based on a measure that we later describe, and the anomalous areas beyond some pre-determined threshold are suggested for further physical tests. We next present the algorithm in Algorithm 2 and then describe its actions.

Algorithm 2 Market-Clustering

$\mathscr{A}$: for $i \in \mathscr{I}$ that participated in some auction, stores the highest bid for i from some cluster
s: anomaly scores based on spatial and shape anomaly measures

---

Procedure MARKETCLUSTERING $(k, scores)$

8   $clusters = \text{INITCLUSTERS}(k, scores)$;
9   $status = $ not converged ;
10 **while** $status \neq converged$ **do**
11     $status = $ converged ;
12   sort $clusters$ by their $wallet$;
13   **for** $c \in clusters$ **do**
14        $(i, o) \leftarrow$ cheapest couple, $i \in c.\mathscr{I}$, $o \notin c.\mathscr{I}$ based on PRICE(i, o) ;
15   **if** $o \in \mathscr{A}$ and $c.\mathscr{W} \leq \mathscr{A}[o]$ **then**
16   **goto** 14 ;
17        $status = \text{ATTEMPTTOEXPAND}(c, i, o, \mathscr{A})$;
18   **if** $status = merged$ **then**
19   **goto** 10 ;
20 **return** $clusters$;

Procedure INITCLUSTERS (k, scores)

21 **for** $i \in [1, k]$ **do**
22     $core = $ imp' with the i-highest anomaly score ;
23   c: New Cluster ;
24     $c.\mathscr{C} = [coreImpurity]$;   // core imps'
25     $c.\mathscr{I} = [coreImpurity]$;   // imps' inside
26     $c.\mathscr{W} = \text{F}(scores[coreImpurity])$;
27     $clusters.\text{APPEND}(c)$;
28 **return** $clusters$;

Procedure ATTEMPTTOEXPAND $(c, i, o, \mathscr{A})$

29 **if** $\exists c' \in clusters$ s.t. $o \in c'.\mathscr{C}$ **then**
30     $\mathscr{A}[o] = c.\mathscr{W}$;   // place bid
31     $c.\mathscr{W} = c.\mathscr{W} + c'.\mathscr{W}$ $c.\mathscr{C} = \text{CONCATENATE}(c.\mathscr{C}, c'.\mathscr{C})$;
32     $c.\mathscr{I} = \text{CONCATENATE}(c.\mathscr{I}, c'.\mathscr{I})$;
33     $clusters.\text{REMOVE}(c')$;
34   **return** merged ;
35 **else if** $c.\mathscr{W} \geq \text{PRICE}(i, o)$ **then**
36     $\mathscr{A}[o] = c.\mathscr{W}$;   // place bid
37     $c.\mathscr{W} = c.\mathscr{W} - \text{PRICE}(i, o)$;   // pay
38     $c.\mathscr{I}.\text{APPEND}(o)$;
39   **if** $\exists c' \in clusters$ s.t. $o \in c'.\mathscr{I}$ **then**
40        $c'.\mathscr{I}.\text{REMOVE}(o)$;
41 **return** converged ;

Procedure PRICE (i, o)

42 $d = \left(\exp \left(\sqrt{\text{IMPURITYDIST}(i, o)} \right) \right)^{c_1}$;
43 $s = \left(1 - \left(s[i] * c_2^1 \right)^{c_3^1} * \left(s[o] * c_2^2 \right)^{c_3^2} \right)^{c_4}$;
44 $price = d * s$;
45 **if** $\exists c' \in clusters$ s.t. $o \in c'.\mathscr{C}$ **then**
46     $dis = \left(1 - \left(s[i] * c_2^3 \right)^{c_5^1} * \left(s[o] * c_2^4 \right)^{c_5^2} \right)^{c_6}$;
47     $pen = \left(2 - |s[i] - s[o]| \right)^{c_7}$;
48     $price = price * dis * pen$;
49 **return** $price$;

The MARKETCLUSTERING procedure is the main procedure of the algorithm. It receives as input a parameter k—number of initial clusters, and a list *scores*—impurities' anomaly scores, which in our case are the combination of the spatial and shape measures. First we initialize the list *clusters* in line 8, by defining in INITCLUSTERS for each cluster c its *Core impurities* list—$c.\mathscr{C}$, *Impurities inside* list—$c.\mathscr{I}$, and *Wallet balance* variable—$c.\mathscr{W}$. The procedure in lines 21–28 does that by setting the core impurities of k clusters with the k most anomalous impurities (lines 24, 25). Core impurities represent each cluster; thus, the degree of the anomaly of each cluster is first defined by the anomaly score of the initial core impurity, and it is stored in the wallet of the cluster in line 26. The loop in line 10 iterates until convergence, and in each iteration, it sorts the clusters by their wallet and initiates the main loop in line 13. This loop iterates over all clusters, and for each cluster c it finds in line 14 the cheapest couple (i, o), s.t. $i \in c.\mathscr{I}$ is an impurity inside of it, and $o \in \mathscr{I} \setminus c.\mathscr{I}$ is an impurity outside of it, under some parametric price function PRICE(i, o). This couple, in fact, implies that the cheapest impurity for cluster c to append is o, and the price for it is PRICE(i, o). We suggest that the price will be a function of distance and anomaly score, i.e., the price should decrease as the distance decreases to encourage clusters to be continuous, and as the anomaly scores increase to instruct clusters to expand towards anomalous impurities and cover a larger anomalous area. We later present our parametric price function in PRICE procedure. In line 15 we make use of $\mathscr{A}$ which is a list that stores for each impurity $i \in \mathscr{I}$ what is the cluster with the highest wallet balance that tried to append i to himself. In order to prevent clusters from fighting and emptying their wallets over impurities, we allow c to proceed to line 17 only if it is the cluster with the highest balance that has attempted to append o until now. In line 17, c attempts to expand its reach by calling the procedure ATTEMPTTOEXPAND. In line 29 we check if the other impurity o is a core impurity of some other cluster c'. If it is, first $\mathscr{A}$ is updated with the new bid on o, and then the clusters $\{c, c'\}$ are merged into c. Otherwise, as a utilization of *credit with no overdraft* policy, if there is enough credit in the wallet of c, again $\mathscr{A}$ is updated with the new bid on o, and c pays for and appends o. Then we check in line 18 if a merge has occurred, and if it did, we sort the clusters again by their wallet balance

---

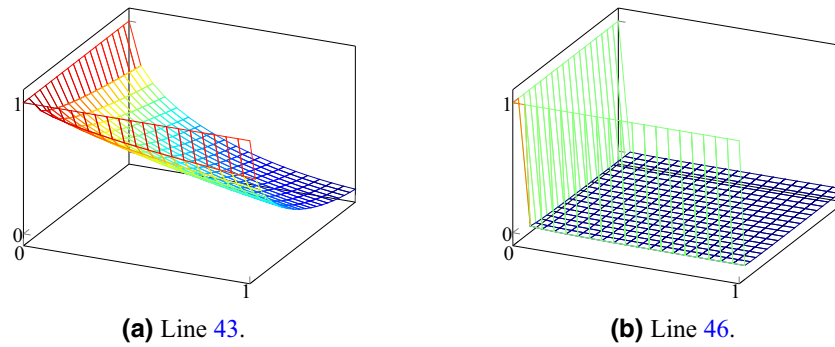(a) Line 43.　　　　　　　　(b) Line 46.

Figure 17. Proposed functions from Algorithm 2.

and proceed to iterate over all new clusters in line 13. The parametric price function that we used in line 14 is presented in PRICE procedure. The parameters that we used are: $c_1 = 1.7$, for $i \in [1, 4]$ $c_2^i = 0.95, c_3^1 = c_3^2 = 0.5$, $c_4 = 1.6, c_5^1 = c_5^2 = 0.05, c_6 = 2.5, c_7 = 8$. The price between two impurities i, o is determined by a function of the distance between them in line 42 and by a function of how anomalous are they in line 43. d grows with the distance, i.e. the price gets higher as the impurities are more distant from each other. However, s scales negatively with the anomaly scores of i and o. The behavior of the function can be seen at Fig. 17a. Since we want clusters to merge (line 29) and span a larger anomalous area, we check in line 45 if o is a core impurity of some cluster, and if it is, there is a price reduction in line 46. The value *dis* falls much more drastically than s. This is because we encourage clusters merging, therefore, giving a relatively low price to core impurities despite the distance to them. The behavior of the function in line 46 is presented in Fig. 17b. Line 47 penalizes cluster merging of similar sizes in order to encourage big and anomalous clusters to absorb smaller and less anomalous clusters at a lower price.

After marking the anomalous areas, we now quantify the degree of the anomaly of each area. First, we suggest the following area anomaly measure for cluster c, and we next prove that it indeed indicates how c is anomalous.

Theorem 1 *am(c) is monotonically increasing with the degree of anomaly of c, based on Market-Clustering algorithm and on c's Spatial and Shape anomaly score, where*

$$am(c) := \sum_{i \in c.\mathscr{I}} \left(Score(i) * Area^2(i) \right) * Diameter(c) * |c.\mathscr{I}| \tag{2}$$

Proof Appending lots of non-anomalous (and not core) impurities is an expensive procedure, compared to appending lots of anomalous impurities because of the discount in line 43 for anomalous impurities. Thus, clusters with a large number of impurities apparently have included cheaper, more anomalous impurities. Moreover, clusters that have appended core impurities, which are the most anomalous, clearly should be considered as more anomalous. Indeed, cluster merging imposes a higher wallet balance for future impurities addition, in addition to a concatenation of impurities in each cluster. Thus, as the number of impurities in the cluster, $|c.\mathscr{I}|$, grows, the degree of the anomaly of c grows correspondingly. Similarly, appending some far impurity o (line 42) is naturally an expensive operation, as long as o is not anomalous (line 43). Thus, if a cluster overcame the expenses of appending distant impurities, it is because it has presumably appended anomalous impurities. Therefore, as $Diameter(c)$ grows, the degree of the anomaly of c grows as well. Finally, big and anomalous impurities highly imply that the cluster has a high spatial anomaly score. Therefore the component $\sum_{i \in c.\mathscr{I}} \left(Score(i) * Area^2(i) \right)$ grows with the degree of anomaly of c. Since all components are monotonically increasing with the degree of the anomaly of c, and because multiplication preserves monotonicity, $am(c)$ is monotonically increasing with the degree of the anomaly of c. □

We also note that similarly to all other anomaly measures presented in this work, the usage of multiplication enhances the anomaly scores of clusters with high scores in each of the components, compared to clusters with lower scores on some of the components. We present the output of the algorithm ($k = 10$), on top of spatial and shape anomaly measures, after ordering the clusters based on Eq. (2), in Fig. 16f. The most anomalous cluster is the red one.

Anomaly detection evaluation—tuning via feedback

The promise of an end-to-end model for a diverse field such as QM is only viable in the presence of tuning via feedback. For the task of data segmentation, there is a clear link between the number of labeled data pieces and the model's accuracy. As previously stated, unlike evaluation benchmarks for typical machine learning tasks, in this work, neither the data nor the labeling is trivial, as the data is not natural, extremely unique (even in the electron microscopy arena[58]), and inherently problematic to classify binary. As a result, in case of exceptional data to the model or indecisive outcomes, transferring the current model understating should be relatively easy using more labeled data. However, for the anomaly detection task, there are no specific mathematical guidelines for determining anomaly in the aspects of QM a priori (even in cases that benchmarks are present, an anomaly
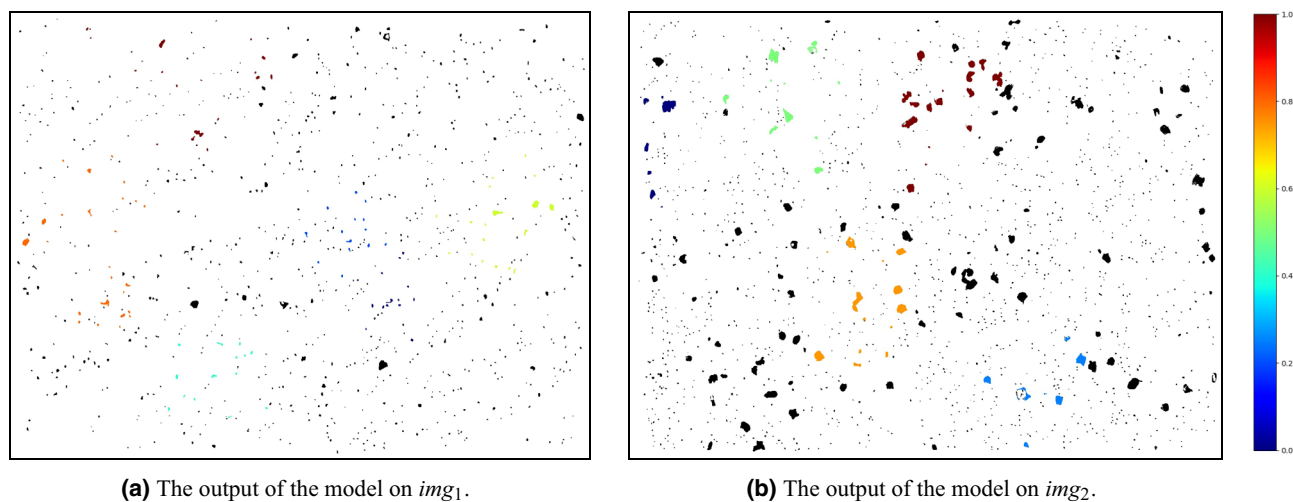
(a) The output of the model on img_1.

(b) The output of the model on img_2.

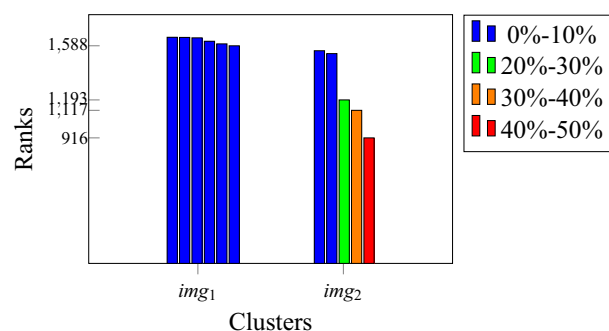**Figure 18.** Anomaly detection pipeline applied on a test sample.

**Figure 19.** Ranks of all clusters. Each group is a single test scan.

is still subjected to perspective[93]). Moreover, from the physical perspective, and even in the presence of big data, there is no guarantee that a mathematical *relative* anomaly is also a *actual physical* anomaly, even when the said model takes into consideration all of the relevant properties of the material. Furthermore, different materials can result in different mathematical and physical anomaly thresholds, and as such, for each case, there is a need to ratify the model results by experiments, or rather re-tune it via experimental feedback. As a result, we introduce a physical method for feedback and re-tune the anomaly detection model to improve our big-data analysis.

We suggest testing the anomaly detection model with the following procedure: prepare fresh metallographic samples and use the model to locate and quantify the anomaly scores of the most anomalous areas of impurities inside them. The model outputs are then used in order to determine whether and where there were physical defects in the materials, specifically in the areas of interest (see "Anomaly detection measures"). For example, the results of two samples are shown in Fig. 18a,b. The most anomalous test scans areas were ordered together with all other scans in the dataset, and they were placed in places 1588 and 916, respectively, out of a total of 1653 clusters. Cluster 1588 resulted under the first decile, while 916 was placed between the fourth and fifth deciles. All the results are shown in Fig. 19.

After getting the outputs from the model, two examinations should be made: (1) Microhardness Vickers (MHV)[94] test in the vicinity of the most anomalous inclusions (the inclusions size and the microhardness trace are both on the same scale of few to tens micrometers) and in normal areas; (2) EDS (energy dispersive spectroscopy) analysis in a SEM[95] to evaluate the inclusions composition. For example, for the two said samples, we performed an examination by MHV and EDS to determine whether the mathematical *relative* anomaly is also a *actual physical* anomaly and found in this case that there was no difference between normal and anomalous inclusions. Although it is clear that the most anomalous area in Fig. 18b looks much more anomalous than the others. Since we observed that there is no difference between normal and anomalous areas, we conclude that while our mathematical *relative* anomaly detection model is capable of quantifying successfully how each area of inclusions is anomalous compared to all other areas, it is the task of the experienced user of the model to determine the threshold from which the area is considered anomalous enough to be defective and to re-tune the mathematical model accordingly.

Appendix 1: Data examples
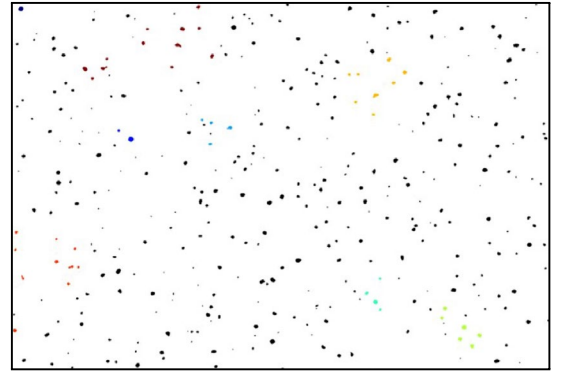See Figs. 20, 21, 22, 23, 24 and 25.
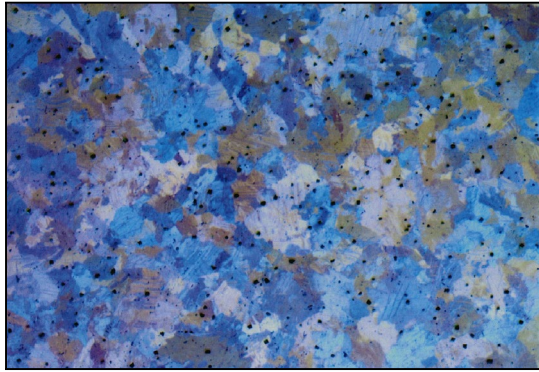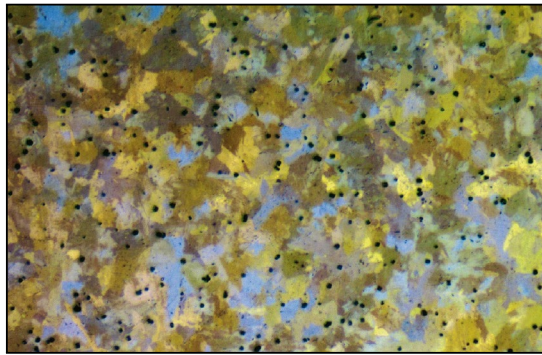
Input image

Impurities segmentation
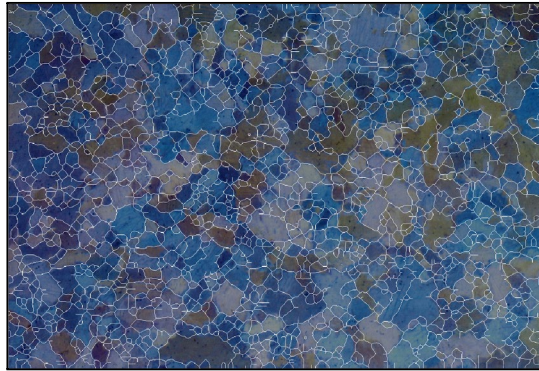
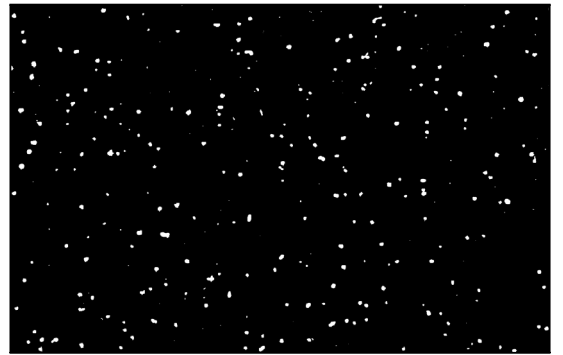Inpainting and GB segmentation

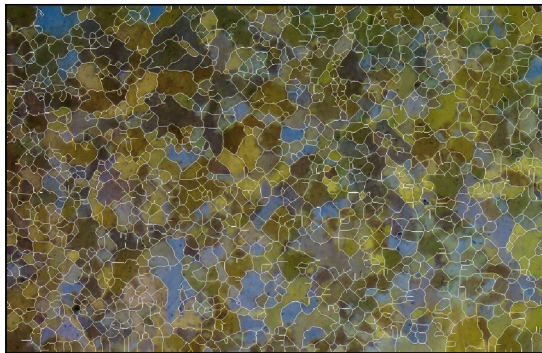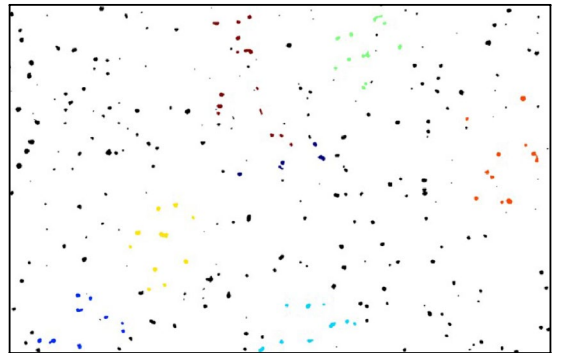
Area anomaly detection

Figure 20. Example #1.

Input image
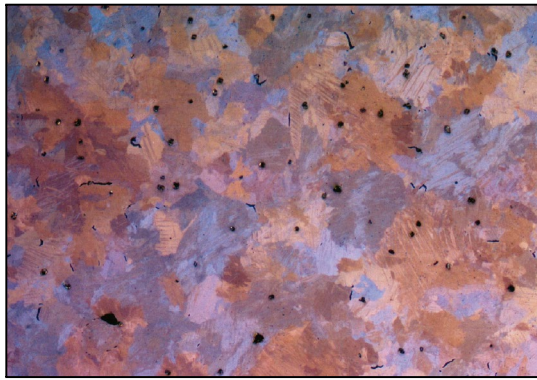

Impurities segmentation
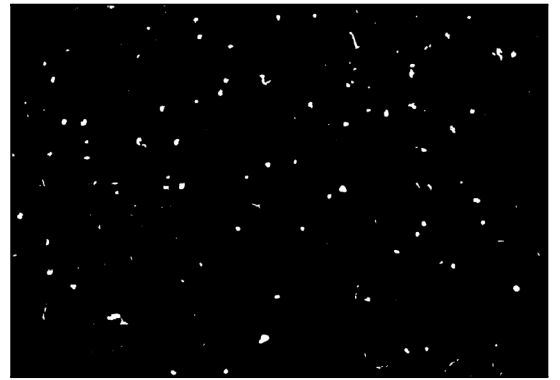

Inpainting and GB segmentation
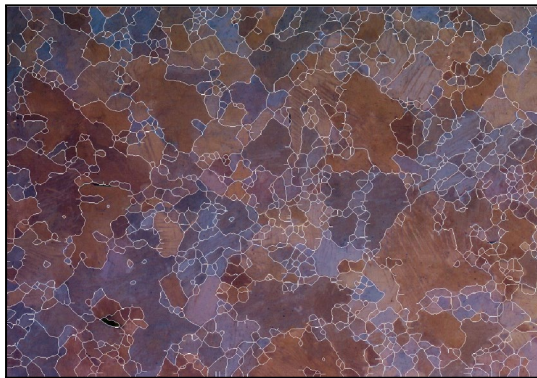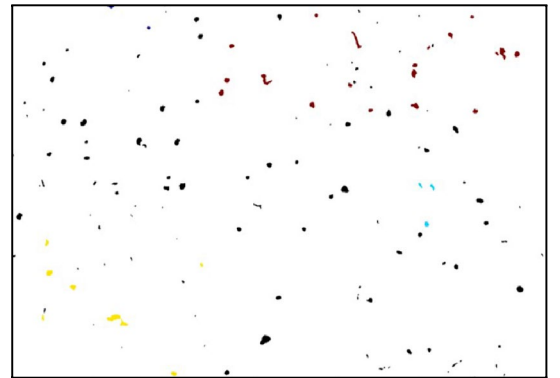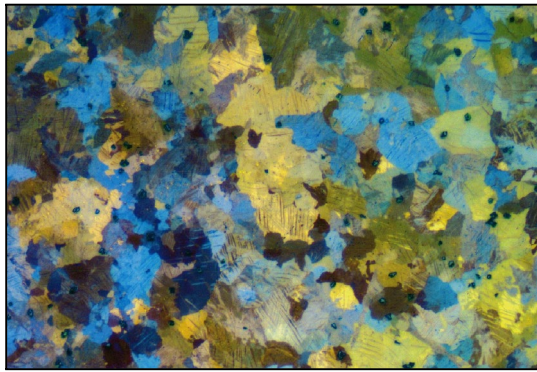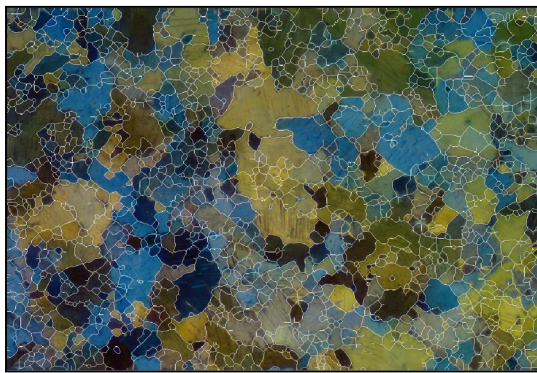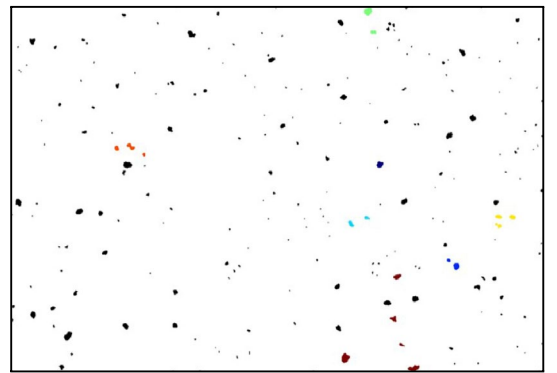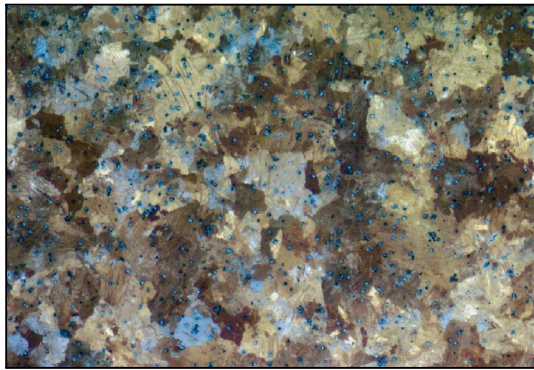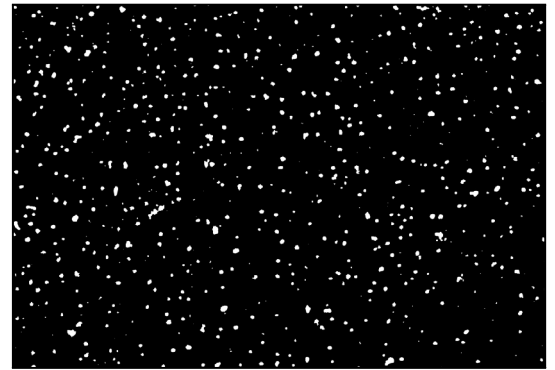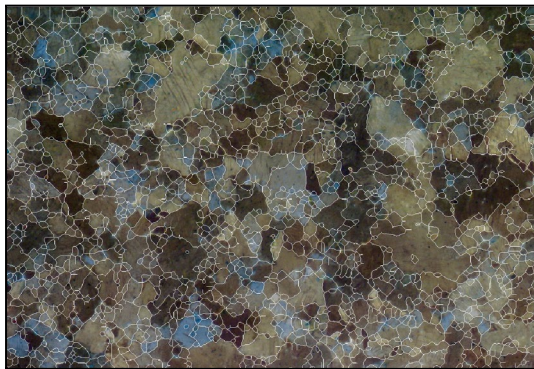

Area anomaly detection

Figure 21. Example #2.

Input image

Impurities segmentation

Inpainting and GB segmentation

Area anomaly detection

Figure 22. Example #3.

Input image

Impurities segmentation

Inpainting and GB segmentation

Area anomaly detection

Figure 23. Example #4.

Input image

Impurities segmentation

Inpainting and GB segmentation
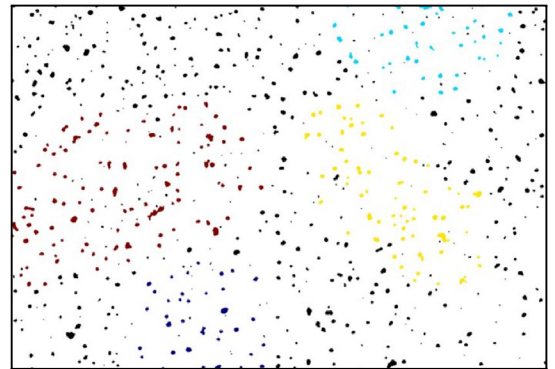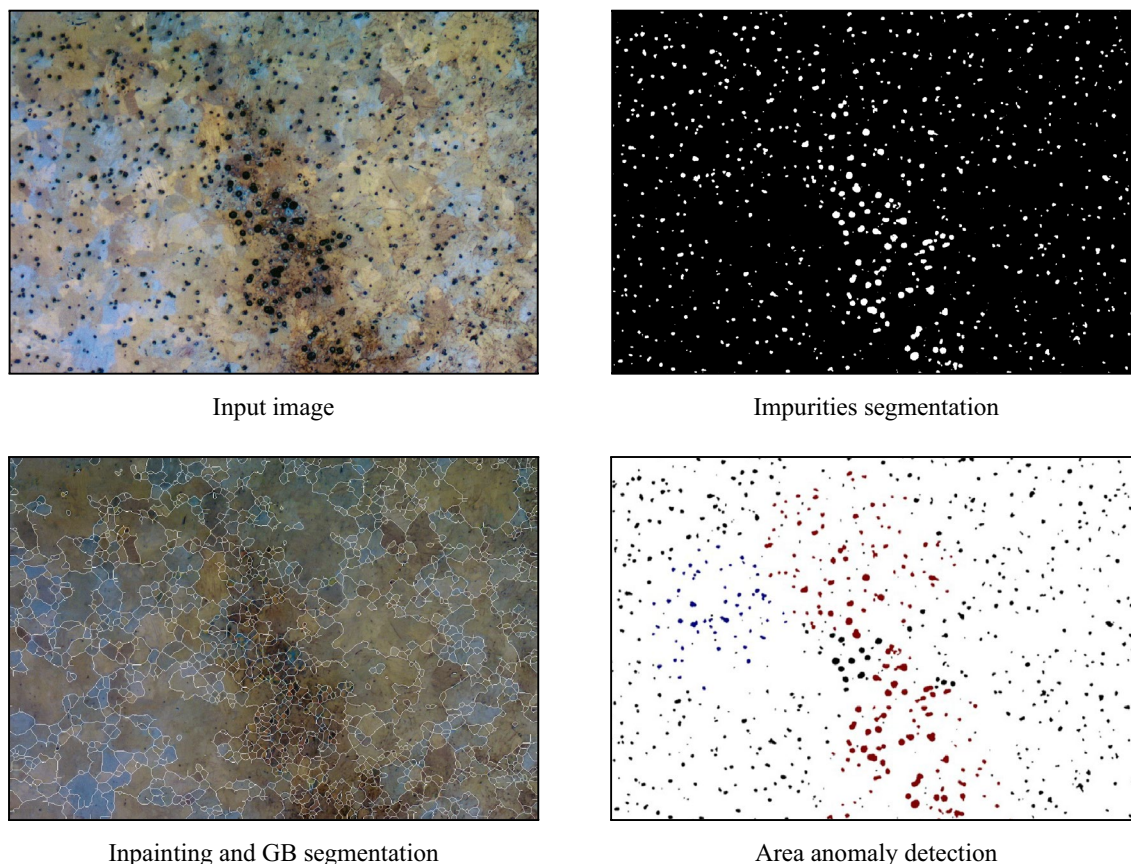

Area anomaly detection

Figure 24. Example #5.

Input image

Impurities segmentation

Inpainting and GB segmentation

Area anomaly detection

Figure 25. Example #6.

References

1. Sinha, A. K. *Physical Metallurgy Handbook* (McGraw-Hill Professional Publishing, 2003).
2. Wang, N., Wang, Z., Aust, K. & Erb, U. Effect of grain size on mechanical properties of nanocrystalline materials. *Acta Metall. Mater.* **43**, 519–528 (1995).
3. Naghizadeh, M. & Mirzadeh, H. Effects of grain size on mechanical properties and work-hardening behavior of AISI 304 austenitic stainless steel. *Steel Res. Int.* **90**, 1900153 (2019).
4. Armstrong, R. The influence of polycrystal grain size on several mechanical properties of materials. *Metallurg. Mater. Trans. B* **1**, 1169–1176 (1970).
5. Ralston, K. & Birbilis, N. Effect of grain size on corrosion: A review. *Corrosion* **66**, 075005 (2010).
6. Ralston, K., Fabijanic, D. & Birbilis, N. Effect of grain size on corrosion of high purity aluminium. *Electrochim. Acta* **56**, 1729–1736 (2011).
7. Brunner, J., Birbilis, N., Ralston, K. & Virtanen, S. Impact of ultrafine-grained microstructure on the corrosion of aluminium alloy aa2024. *Corros. Sci.* **57**, 209–214 (2012).
8. Zeng, H. *et al.* Grain size-dependent electrical resistivity of bulk nanocrystalline gd metals. *Progress Nat. Sci. Mater. Int.* **23**, 18–22 (2013).
9. Ivanov, O., Maradudina, O. & Lyubushkin, R. Grain size effect on electrical resistivity of bulk nanograined bi2te3 material. *Mater. Charact.* **99**, 175–179 (2015).
10. Andrews, P., West, M. & Robeson, C. The effect of grain boundaries on the electrical resistivity of polycrystalline copper and aluminium. *Philos. Mag.* **19**, 887–898 (1969).
11. Askeland, D. R. & Phulé, P. P. *The Science and Engineering of Materials* (1994).
12. Liu, M. *et al.* Normal hall-petch behavior of mild steel with submicron grains. *Mater. Lett.* **57**, 2798–2802 (2003).
13. Volpp, T., Göring, E., Kuschke, W.-M. & Arzt, E. Grain size determination and limits to hall-petch behavior in nanocrystalline nial powders. *Nanostruct. Mater.* **8**, 855–865 (1997).
14. Naik, S. N. & Walley, S. M. The hall-petch and inverse hall-petch relations and the hardness of nanocrystalline metals. *J. Mater. Sci.* **55**, 2661–2681 (2020).
15. Wu, C.-Z. & Shih, T.-S. Effects of inclusion particles on the microstructure and mechanical properties of high strength austempered ductile iron. *Mater. Trans.* **44**, 995–1003 (2003).
16. Thornton, P. The influence of nonmetallic inclusions on the mechanical properties of steel: A review. *J. Mater. Sci.* **6**, 347–356 (1971).
17. Meurling, F., Melander, A., Tidesten, M. & Westin, L. Influence of carbide and inclusion contents on the fatigue properties of high speed steels and tool steels. *Int. J. Fatigue* **23**, 215–224 (2001).
18. Rankine, W. J. M. On the causes of the unexpected breakage of the journals of railway axles; and on the means of preventing such accidents by observing the law of continuity in their construction. In *Minutes of the Proceedings of the Institution of Civil Engineers*, vol. 2, 105–107 (Thomas Telford-ICE Virtual Library, 1843).

19. Braithwaite, F. On the fatigue and consequent fracture of metals. In *Minutes of the Proceedings of the Institution of Civil Engineers*, vol. 13, 463–467 (Thomas Telford-ICE Virtual Library, 1854).
20. Bagnoli, F. & Bernabei, M. Fatigue analysis of a p180 aircraft main landing gear wheel flange. *Eng. Fail. Anal.* **15**, 654–665 (2008).
21. Franco, L. *et al.* Fatigue fracture of a nose landing gear in a military transport aircraft. *Eng. Fail. Anal.* **13**, 474–479 (2006).
22. E 3-11: Standard guide for preparation of metallographic specimens. *ASTM International* (2011).
23. E 112-13: Standard test methods for determining average grain size. *ASTM International* (2013).
24. E 45-97: Standard test methods for determining the inclusion content of steel. *ASTM International* (1997).
25. Iskakov, A. & Kalidindi, S. R. A framework for the systematic design of segmentation workflows. *Integrat. Mater. Manuf. Innov.* **20**, 1–19 (2020).
26. Duval, L. *et al.* Image processing for materials characterization: Issues, challenges and opportunities. In *2014 IEEE International Conference on Image Processing (ICIP)*, 4862–4866 (IEEE, 2014).
27. Ramprasad, R., Batra, R., Pilania, G., Mannodi-Kanakkithodi, A. & Kim, C. Machine learning in materials informatics: Recent applications and prospects. *NPJ Comput. Mater.* **3**, 1–13 (2017).
28. Dimiduk, D. M., Holm, E. A. & Niezgoda, S. R. Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering. *Integrat. Mater. Manuf. Innov.* **7**, 157–172 (2018).
29. DeCost, B. *et al.* Scientific AI in materials science: A path to a sustainable and scalable paradigm. *Mach. Learn. Sci. Technol.* **20**, 20 (2020).
30. Ge, M., Su, F., Zhao, Z. & Su, D. Deep learning analysis on microscopic imaging in materials science. *Mater. Today Nano* **210**, 100087 (2020).
31. Kesireddy, A. & McCaslin, S. Application of image processing techniques to the identification of phases in steel metallographic specimens. In *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering* 425–430 (Springer, 2015).
32. Xing, F., Xie, Y., Su, H., Liu, F. & Yang, L. Deep learning in microscopy image analysis: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 4550–4568 (2017).
33. Moen, E. *et al.* Deep learning for cellular image analysis. *Nat. Methods* **20**, 1–14 (2019).
34. Han, Y., Lai, C., Wang, B. & Gu, H. Segmenting images with complex textures by using hybrid algorithm. *J. Electron. Imaging* **28**, 013030 (2019).
35. Chen, Y. & Chen, J. A watershed segmentation algorithm based on ridge detection and rapid region merging. In *2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 420–424 (IEEE, 2014).
36. Zhang, S., Chen, D., Liu, S., Zhang, P. & Zhao, W. Aluminum alloy microstructural segmentation method based on simple noniterative clustering and adaptive density-based spatial clustering of applications with noise. *J. Electron. Imaging* **28**, 033035 (2019).
37. Liu, J. & Chen, J. An improved iterative watershed according to ridge detection for segmentation of metallographic image. *Metallogr. Image* **8**, 20 (2012).
38. Peregrina-Barreto, H. *et al.* Automatic grain size determination in microstructures using image processing. *Measurement* **46**, 249–258 (2013).
39. Marin, F. B., Potecaşu, F., Marin, M. & Alexandru, P. Automatic grain size determination in microstructures using computer vision algorithm based on support vector machine (SVM). In *Advanced Materials Research* Vol. 1143 194–199 (Trans Tech Publ, 2017).
40. Wu, W.-H., Lee, J.-C. & Wang, Y.-M. A study of defect detection techniques for metallographic images. *Sensors* **20**, 5593 (2020).
41. For Testing, A. S. & Materials. E112-13: Standard test methods for determining average grain size (2013).
42. ISO, I. 643: 2012; Steels-micrographic determination of the apparent grain size (2012).
43. International, A. E1382-97. Standard test methods for determining average grain size using semiautomatic and automatic image analysis (2015).
44. DeCost, B. L. & Holm, E. A. A computer vision approach for automated analysis and classification of microstructural image data. *Comput. Mater. Sci.* **110**, 126–133 (2015).
45. Azimi, S. M., Britz, D., Engstler, M., Fritz, M. & Mücklich, F. Advanced steel microstructural classification by deep learning methods. *Sci. Rep.* **8**, 1–14 (2018).
46. DeCost, B. L., Lei, B., Francis, T. & Holm, E. A. High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel. arXiv:1805.08693 (arXiv preprint) (2018).
47. Ma, B. *et al.* Deep learning based automatic inpainting for material microscopic images. *J. Microsc.* **20**, 20 (2020).
48. Kondo, R., Yamakawa, S., Masuoka, Y., Tajima, S. & Asahi, R. Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics. *Acta Mater.* **141**, 29–38 (2017).
49. Underwood, E. E. Quantitative metallography. *ASM Int. ASM Handb.* **9**, 123–134 (1985).
50. Habeeb, R. A. A. *et al.* Real-time big data processing for anomaly detection: A survey. *Int. J. Inf. Manage.* **45**, 289–307 (2019).
51. Lehto, P., Remes, H., Saukkonen, T., Hänninen, H. & Romanoff, J. Influence of grain size distribution on the hall-petch relationship of welded structural steel. *Mater. Sci. Eng. A* **592**, 28–39 (2014).
52. Hall, E. Variation of hardness of metals with grain size. *Nature* **173**, 948–949 (1954).
53. Vander Voort George, F. *Metallographic: Principles and Practice* (McGraw-Hill, 1984).
54. Beeley, P. *Foundry Technology* (Elsevier, 2001).
55. ASM International. In *Materials Characterization* (ed. Sag, D.) (ASM International, 2019). https://doi.org/10.31399/asm.hb.v10.9781627082136.
56. Zhang, X.-Y., Liu, C.-L. & Suen, C. Y. Towards robust pattern recognition: A review. *Proc. IEEE* **108**, 894–922 (2020).
57. Morgan, D. *et al.* Machine learning in nuclear materials research. *Curr. Opin. Solid State Mater. Sci.* **26**, 100975 (2022).
58. Ede, J. M. Deep learning in electron microscopy. *Mach. Learn. Sci. Technol.* **2**, 011004 (2021).
59. Holm, E. A. *et al.* Overview: Computer vision and machine learning for microstructural characterization and analysis. *Metallurg. Mater. Trans. A* **51**, 5985–5999 (2020).
60. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241 (Springer, 2015).
61. Burke, J., Colling, D., Gorum, A. & Greenspan, J. Physical metallurgy of uranium alloys. Tech. Rep. (Brook Hill Publishing Co., Chestnut Hill, MA, 1976).
62. DeCost, B. L. *et al.* Uhcsdb: Ultrahigh carbon steel micrograph database. *Integrat. Mater. Manuf. Innov.* **6**, 197–205 (2017).
63. Rusanovsky, M., Beeri, O. & Oren, G. Mlography: Impurities tags data-set. https://github.com/Scientific-Computing-Lab-NRCN/MLography/tree/master/tags_png_cropped (2021). [Online].
64. Rusanovsky, M., Beeri, O. & Oren, G. Mlography: Impurities segmentation images. https://github.com/Scientific-Computing-Lab-NRCN/MLography/tree/master/Segmentation/unet/data/small/train (2021). [Online].
65. Rusanovsky, M., Beeri, O. & Oren, G. Mlography: Grains boundary segmentation images. https://github.com/Scientific-Computing-Lab-NRCN/MLography/tree/master/Segmentation/unet/data/squares_128/train (2021). [Online].
66. Rusanovsky, M., Beeri, O. & Oren, G. Mlography: Big metallography images. https://github.com/Scientific-Computing-Lab-NRCN/MLography/tree/master/Segmentation/unet/data/metallography/train/image (2021). [Online].
67. Rusanovsky, M., Beeri, O. & Oren, G. Mlography: Fully segmented images. https://github.com/Scientific-Computing-Lab-NRCN/MLography/tree/master/Segmentation/unet/data/post_segmented_edges_binary/masked (2021). [Online].
68. Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440 (2015).
69. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (arXiv preprint) (2014).

70. Iglovikov, V. & Shvets, A. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. arXiv:1801.05746 (arXiv preprint) (2018).
71. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988 (2017).
72. Maninis, K.-K., Caelles, S., Pont-Tuset, J. & Van Gool, L. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 616–625 (2018).
73. Papadopoulos, D. P., Uijlings, J. R., Keller, F. & Ferrari, V. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE International Conference on Computer Vision*, 4930–4939 (2017).
74. Liu, G. *et al.* Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 85–100 (2018).
75. Yu, J. *et al.* Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5505–5514 (2018).
76. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 679–698 (1986).
77. Xia, X. & Kulis, B. W-net: A deep model for fully unsupervised image segmentation. arXiv:1711.08506 (arXiv preprint) (2017).
78. Poma, X. S., Riba, E. & Sappa, A. Dense extreme inception network: Towards a robust CNN model for edge detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1923–1932 (2020).
79. Guo, Z. & Hall, R. W. Parallel thinning with two-subiteration algorithms. *Commun. ACM* **32**, 359–373 (1989).
80. Beucher, S. Use of watersheds in contour detection. In *Proceedings of the International Workshop on Image Processing* (CCETT, 1979).
81. Plus, I.-P. v. 6.0, Media Cybernetics Inc, Rockville, Md, USA; 2018.
82. Goldstein, M. & Uchida, S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS One* **11**, e0152173 (2016).
83. Knox, E. M. & Ng, R. T. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases*, 392–403 (Citeseer, 1998).
84. Ramaswamy, S., Rastogi, R. & Shim, K. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 427–438 (2000).
85. Zhang, K., Hutter, M. & Jin, H. A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 813–822 (Springer, 2009).
86. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**, 21–27 (1967).
87. Igathinathane, C., Pordesimo, L., Columbus, E., Batchelor, W. & Methuku, S. Shape identification and particles size distribution from basic shape parameters using ImageJ. *Comput. Electron. Agric.* **63**, 168–182 (2008).
88. Hawkins, S., He, H., Williams, G. & Baxter, R. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, 170–180 (Springer, 2002).
89. Dau, H. A., Ciesielski, V. & Song, A. Anomaly detection using replicator neural networks trained on examples of one class. In *Asia-Pacific Conference on Simulated Evolution and Learning*, 311–322 (Springer, 2014).
90. NegevHPC Project. www.negevhpc.com. [Online].
91. Abadi, M. *et al.* Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283 (2016).
92. Deepak, P. Anomaly detection for data with spatial attributes. In *Unsupervised Learning Algorithms* 1–32 (Springer, 2016).
93. Chalapathy, R. & Chawla, S. Deep learning for anomaly detection: A survey. arXiv:1901.03407 (arXiv preprint) (2019).
94. Bückle, H. Progress in micro-indentation hardness testing. *Metallurg. Rev.* **4**, 49–100 (1959).
95. Goldstein, J. I. *et al. Scanning Electron Microscopy and X-ray Microanalysis* (Springer, 2017).

Acknowledgements

Author contributions

This research was carried out under the supervision of G.O. M.R and G.O wrote the main manuscript text and O.B formulated the materials science background. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to G.O.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.