A Clustering Based Image Segmentation Procedure to Automatically Detect Grains in Polycrystalline Materials

Dženana Alagić and Jürgen Pilz

Abstract

The physical and mechanical properties of a polycrystalline material depend on its microstructure characteristics such as the size and morphology of grains. In practice, different imaging methods are used to visualize the grain structure of such materials. To analyze microstructural changes in case of applied stress and to predict its performance in a given application, the quantitative information about the grain structure must be taken into account. In this work, an efficient and reproducible algorithm, which automatically detects grains in different types of microstructure images, is proposed. Due to the diversity between the analyzed images and a limited number of labeled data, a clustering patch-based approach is followed. The algorithm aims to distinguish between patches in homogeneous grain areas and those lying on grain boundaries through Gaussian Mixture Modeling. The identified groups of grain patches are used to create the seed image for a Seeded Region Growing algorithm, enabling finally a pixelwise image segmentation.

Keywords: Microstructure; Grain segmentation; Gaussian Mixture Modeling; Seeded Region Growing; Patch-based clustering

1 Introduction

Polycrystalline materials are solids composed of a large number of grains of different size and orientation. Grain boundaries refer to the area where two neighboring grains with different orientations meet. It is well known that the microstructural characteristics of polycrystalline materials such as the size and morphology of grains, grain boundaries’ length etc., highly influence the physical and mechanical properties of a material. For example, the strength of a material increases by reducing the grain size [1, 2]. Therefore, the ability to characterize the microstructure is of utmost importance in research studies, as it enables to anticipate the performance of a material in a given application. For this purpose, quantitative analysis of microstructural changes that occur as a consequence of different varying parameters such as heating temperature, is necessary.

In a material science laboratory, a set of tools for characterizing the microstructure of materials is used. One of the most powerful tools in this area of research is Electron Backscattered Diffraction (EBSD), which enables to extract information about grain geometry and grain orientation from the surface area of a material. The drawback of this technique is that it is expensive and time-consuming. Although scientists have made some progress towards reducing the measurement time, the
specimen preparation and data processing are still very demanding, motivating the search for an alternative method. Scanning Electron Microscopy (SEM) and Focused Ion Beam (FIB) are commonly used tools to visualize the grain structure of a material with a fast processing time. However, SEM and FIB provide only visual output, without any additional quantitative information about the grain orientation and structure. To automatically extract as much information as possible out of these images as well, it is necessary to use image processing and statistical methods. In this work, an algorithm is proposed to automatically perform grain segmentation in different types of microstructure images, such as SEM and FIB, and to provide quantitative data for further analysis.

2 Methodology

The variation in the quality of the investigated images is high. It is influenced by various experimental factors including the specimen’s preparation process, the image acquisition process, where different (detector) settings are possible, and the operator, i.e., a human factor is involved, which cannot be neglected. Due to a limited number of ground truths and a very complicated process of producing them, an unsupervised approach is followed. The microstructure images share an important common property: the grain areas are homogeneous and separated by grain boundaries. The method that shows promising results with this type of images is a region-based algorithm called Seeded Region Growing (SRG). However, the automatic selection of seed points is a big challenge. For this task, we propose a workflow illustrated in Fig. 1. The SRG algorithm and the whole workflow are explained in more detail below.

2.1 Seeded Region Growing Algorithm

SRG is an algorithm that is widely used for semantic image segmentation. It is highly effective for images in which pixels belonging to the same region have the same or similar intensity values [3]. The algorithm requires a seed point for each region in the image. To avoid the possible outlier effect, it is recommended to use small areas as seeds instead of single pixels. Based on seeds, the initial mean gray value for each region is computed. Therefore, each seed area should be large enough to provide a stable estimate of the region’s mean. This value is compared with the pixels in the seed’s neighborhood. At each step of the algorithm, the pixel with the smallest difference to a neighboring region is added to that region and the region’s mean is updated. All other neighboring pixels that are not yet assigned to any region are added to a candidate list. The number of seeds equals the number of regions to be detected. If some region does not have a starting seed, two possible outcomes may occur. Either this region will be completely contained in some of the neighboring regions or it will be partitioned between two or more neighboring regions. The first case is a typical example of an undersegmentation, which can be solved in a post-processing step via splitting methods. The second case mentioned above is worse, as it leads to a wrong detection also of those regions for which the seeds were perfectly defined. Thus, the appropriate seed selection is crucial for a good algorithm performance.

In general, the SRG algorithm is fast, robust and free of tuning parameters. However, the automatic seed selection is still a big challenge. A lot of solutions were
proposed over the years, but the problem has not yet been solved completely [4, 5]. Most of the proposed solutions assume that the regions in the image have different intensity values or the number of regions is known in advance. In our application, the regions, i.e. the grain areas in the microstructure images, may have similar mean gray values and their number is unknown. The grain boundaries are the only property that enable to distinguish between neighboring grains. To automatically select seeds in microstructure images, a patch-based approach is proposed. More precisely, the image is divided into small non-overlapping rectangular selections called patches. The goal is to distinguish between patches that lie completely inside grains and the ones lying on a grain boundary. To do this, features that measure the homogeneity of areas are computed for each patch. Based on these features, a Gaussian Mixture Model (GMM) is used to divide the patches into two classes: grain and grain boundary area. The connected components of the grain patches are used as seed points. For better understanding, the algorithm steps are illustrated with an example image in Fig. 2.

2.2 Image Denoising

The choice of an appropriate denoising method is a very important step in the algorithm. The output influences not only the patch features but also the performance of the SRG algorithm in the final step. It is hard to define a single method that is suitable for all types of microstructure images since the noise level as well as its distribution varies heavily. In FIB images, a large number of outlier pixels is present in each region. This kind of image distortion can be easily fixed with a smoothing filter. But, the application of a smoothing filter on SEM images leads to information loss since some grain boundaries are not well delineated. In such a case, an edge preserving algorithm is more suitable. Therefore, two denoising methods are considered here, depending on the image type: Gaussian Blur for FIB images and Non-local means (NL-means) for SEM images. The effects of these methods on both image types are shown in Fig. 3.

2.2.1 Gaussian Blur

Gaussian Blur is the most commonly used filter for noise reduction. It blurs the image by convolving with a kernel that corresponds to a discrete, two-dimensional Gaussian function

\[ G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \]

where \( \sigma \) denotes the standard deviation of the Gaussian distribution, \( x \) is the horizontal, and \( y \) the vertical distance from the center pixel. The size of the kernel is determined by \( \sigma \). To remove the noise in FIB images, but preserve the information of grain boundaries, we used \( \sigma = 2 \) [6].

2.2.2 Non-local Means Denoising

NL-means is a denoising method that preserves fine details and textures within an image. For each pixel, the new value is computed as a weighted average of pixels with similar patches (small neighborhoods) within a given search window (large
neighborhood) [7]. More precisely, let $v = \{ v(i) | i \in I \}$ denote the input noisy image. The estimated value for some pixel $i$, is computed as follows:

$$NL[v](i) = \frac{1}{Z(i)} \sum_{j \in \mathcal{N}(i)} w(i, j) v(j),$$

(2)

where $w(i, j)$ are non-negative weights, $Z(i)$ is a normalization constant such that for any pixel $i$, $Z(i) = \sum_{j \in \mathcal{N}(i)} w(i, j)$, and $\mathcal{N}(i)$ is the search window referring to the local neighborhood of $i$ to reduce the computation time. The weight $w(i, j)$ depends on the similarity between two square patches centered, respectively, at pixels $i$ and $j$. It is defined as

$$w(i, j) = g_h \left( \sum_{\delta \in \Delta} G_\sigma(\delta)(v(i + \delta) - v(j + \delta))^2 \right),$$

(3)

where $G_\sigma$ is a Gaussian kernel with variance $\sigma^2$, $g_h$ is a continuous non-increasing function so that $g_h(0) = 1$ and $\lim_{x \to \infty} g_h(x) = 0$, and $\Delta$ is the discrete patch region containing the neighboring sites $\delta$. The parameter $h$ controls the degree of filtering. In the implementation that we use [8], $g_h(x) = \frac{1}{1 + (x^2/h^2)}$ is adopted. Since the pixels outside the search window $\mathcal{N}(i)$ do not influence the value $NL[v](i)$, the image is separated into independent disjoint pieces, allowing their parallel processing [9].

The size of the patch and search window are determined based on the value of $\sigma$. When $\sigma$ increases, a larger patch is needed to make patch comparisons robust enough. In that case, the search window must also be increased to improve the denoising process by finding more similar pixels. Thus, the image resolution must be taken into account. For SEM images, we used $\sigma = 3$.

2.3 Feature extraction

After denoising, a grid is created to divide the image into small rectangular cells (patches). The patch size (width and height) is given in number of pixels and is determined by the user. To choose the optimal patch size, two things must be considered. First, the patches should not be too small. Small patches could be completely contained in a grain boundary leading to a false assumption of homogeneous grain areas and are prone to artifacts’ influence. Second, the patches should not be too big. Big patches could be so positioned that none of them lies completely inside smaller grains. In such a case, there would be no chance to detect seeds properly.

The grain patches are mostly homogeneous with a small variation in gray intensities, while the grain boundary patches show dispersion in gray values. In SEM images, a grain boundary is either brighter or darker than the neighboring regions. In FIB images, the neighboring regions can have different intensities, so the grain boundary is defined by the transition between two homogeneous areas. In both cases, the homogeneity is the main property that should be measured in order to distinguish between the two classes. The selected features focus on that fact and can be divided into three groups:

- First order statistic features. Two measures are used: standard deviation and the difference between the mean and modal gray value. Small standard deviation and difference refer to a homogeneous grain area.
- Sobel operator. For each patch independently, a gradient is computed using the Sobel operator. The mean and standard deviation of the gradient are computed and added to the feature list.
- Textural features. For each patch, a corresponding gray-level co-occurrence matrix (GLCM), describing the distribution of co-occuring pixel values at a given distance \(d\), is computed. For an 8-bit image, GLCM denoted with \(P\) is a \(256 \times 256\) matrix, where the \((i,j)\)th entry \(p(i,j)\) represents the frequency with which two pixels separated by distance \(d\) occur in the image, one with gray value \(i\) and the other with gray value \(j\). In our implementation, the matrix is computed using \(d = 1\) and considering the first neighbor on the right. Based on the matrix, 6 features listed in Table 1 are selected and added to the feature list [10].

| Feature                  | Formula                                                                 |
|--------------------------|--------------------------------------------------------------------------|
| Angular Second Moment    | \(f_1 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)^2\)                    |
| Inverse Difference Moment| \(f_2 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1+(i-j)^2} p(i,j)\)  |
| Contrast                 | \(f_3 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-j)^2 p(i,j)\)              |
| Entropy                  | \(f_4 = -\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \log(p(i,j))\)        |
| Homogeneity              | \(f_5 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{\mu(i,j)}{1+|i-j|}\)   |
| Variance                 | \(f_6 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-\mu)^2 p(i,j), \mu = \frac{\mu_x + \mu_y}{2}\) |

Throughout the table, \(p(i,j)\) is the \((i,j)\)th entry in the normalized GLCM \(P(i,j)/R\). \(R\) is a normalizing constant and \(N_g\) is the number of distinct gray values.

### 2.4 Gaussian Mixture Modeling
Based on the extracted features, a GMM is applied to assign the patches to one of the two classes: grain or grain boundary area. GMMs are soft clustering methods, where each data point has a probability of belonging to each cluster. It is assumed that the data can be described by a mixture of \(K\) Gaussian distributions, where \(K\) is equivalent to the number of clusters. Each component in the mixture, i.e. cluster, is characterized by the mean \(\mu_k\), covariance matrix \(\Sigma_k\) and weight \(\pi_k\) that corresponds to the associated probability in the mixture. Statistically speaking, a GMM is defined by

\[
p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k),
\]
where $\mathcal{N}(\mathbf{x}\,|\mu_k, \Sigma_k)$ represents the Gaussian density of the $k$-th component

$$\mathcal{N}(\mathbf{x}\,|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_k|}} \exp \left( -\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1}(\mathbf{x} - \mu_k) \right).$$

(5)

In (5), $d$ denotes the number of features. The weights $\pi_k$ are also called mixing coefficients and satisfy the following conditions

$$\sum_{k=1}^{K} \pi_k = 1, \quad 0 \leq \pi_k \leq 1.$$  

(6)

To estimate the model’s parameters, expectation maximization (EM) algorithm is used. EM is a numerical technique to compute the maximum likelihood estimator of the data. The algorithm steps are summarized below.

1 Assign initial values for $\mu_k, \Sigma_k$ and $\pi_k$.

2 **E step.** Based on the current parameter values, for each data point $\mathbf{x}_n$, a probability that it belongs to a cluster $C_k$, denoted with $\gamma(\mathbf{z}_{nk})$, is computed

$$\gamma(\mathbf{z}_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \mu_j, \Sigma_j)}.$$  

(7)

3 **M step.** Update the parameters’ values.

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(\mathbf{z}_{nk}) \mathbf{x}_n$$  

(8)

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(\mathbf{z}_{nk}) (\mathbf{x}_n - \mu_k^{\text{new}})(\mathbf{x}_n - \mu_k^{\text{new}})^T$$  

(9)

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$  

(10)

where

$$N_k = \sum_{n=1}^{N} \gamma(\mathbf{z}_{nk}).$$  

(11)

4 Evaluate the log-likelihood

$$\ln p(\mathbf{X} | \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k) \right)$$  

(12)

and check for convergence. The algorithm has converged when the change in the log likelihood function or model parameters is below some threshold. If the convergence criterion is not satisfied, go to step 2 [11].

2.5 Final segmentation

Based on the clustering results, a binary image, representing the patches of the two classes, is created. If the grain boundary is one patch wide, two neighboring seeds
have touching vertices. To separate them, the morphological operation erosion is applied. The connected groups of grain patches gained in such a way are used to create a seed image. Finally, SRG is applied to achieve the pixelwise segmentation.

3 Results and Discussions

The proposed algorithm is implemented in Java as an ImageJ plugin [6, 12, 13]. The only input required from the user is the patch size; the rest is performed fully automatically. Both types of images, FIB and SEM, shown in this paper, are analyzed with patches of 10 × 10 pixels. For the analyzed FIB images, only a qualitative inspection of the results is possible, since the ground truths for these images were not provided. For the SEM images, quantitative results and evaluations are given.

3.1 Results for FIB images

As mentioned in the paragraph above, no ground truth was available for this type of images. Thus, the results of the algorithm are compared with the SRG output obtained after manual seed selections. The visual comparisons are presented in Fig. 4. The original image is shown in the first column. The second column illustrates the clustering output after the erosion step. For easier interpretation of the results, the connected grain patches, i.e. the seeds, are colored in the same color as the corresponding regions in the final segmentation shown in the third column. In the last two columns, the manual seeds and the resulting SRG segmentation are shown.

The obtained results show that the presented method is very powerful. It provides robust estimates of the regions’ seeds and good segmentation results. Almost all patches are correctly identified. That means that the selected features are valid indicators of the regions’ homogeneity. The final output depends only on the image to be analyzed leading to robust and reproducible results. Nevertheless, the accuracy of the final segmentation is highly influenced by the image quality and thus also by the presence of different artifacts. For example, the images in the second and third row contain visible vertical lines that go through several regions, which comes from the specimen preparation process. As a consequence, the overlapping patches are identified as grain boundaries, which leads to a false oversegmentation of the underlying regions. To avoid this type of error, the specimen preparation should be carried out in a more sophisticated way.

By comparing the algorithm results with the SRG output after manual seed selection, it can be noted that in some cases, the proposed automatic seed selection method provided more robust seeds and thus more accurate region detection. Such regions are indicated with the black arrows in the last column in Fig. 4. Both pink regions overflew the grain boundary, while the automatic method correctly detected it.

The erosion operation prevents the merge of two neighboring regions into one. However, this step splits a seed in an elongated region where single grain patches have touching vertices. Thereby, multiple seeds for one region are created what leads to oversegmentation, as can be seen in the first row of the Fig. 4. For comparison, see the triangular brown region on the right image edge.

It can be further noted that some small and thin grains were missed by the automatic procedure. This happened because the grain area was divided between
multiple patches, each of which cuts a part of the boundary. Those areas are visible as black “holes” in the clustering output. Anyhow, the small grains do not change the underlying grain size distribution significantly, which is why they are negligible for further analysis. If this cannot be guaranteed, the patch size must be adapted.

3.2 Results for SEM images

Unlike FIB images, SEM cross section images are much more challenging to analyze. As it can be seen, there is almost no difference in gray intensities between the regions in SEM images. Some grain boundaries are not well delineated and the texture of the grains varies heavily. While some are very smooth with a small standard deviation in gray intensities, others can have rough surfaces increasing the standard deviation significantly. It is very hard to distinguish between patches in such areas and those lying on the boundaries in terms of their features. Fig. 5 illustrates the clustering output with two SEM images after applying NL-means and Gaussian Blur as preprocessing steps. The black areas represent the patches identified as grains. It can be seen that NL-means preserves edge information but at the same time it preserves and even highlights fine texture details. As a consequence, only patches in smooth areas are correctly identified as grain areas, while all patches in rough grains are falsely identified as grain boundary. With Gaussian Blur as a preprocessing step, some edge information is lost, but the seeds for rough areas are defined. Therefore, to get the best possible output, these two results are combined in the following way:

1. Create binary clustering output images for both denoising methods.
2. Subtract the NL-means from Gaussian Blur output.
3. Take the NL-means clustering output and add all grain patches from the difference that do not touch any of the grain patches in NL-means.

The clustering output defined in such a way is used to create the seed image. The final results and the corresponding ground truth boundaries for these two images are shown in Fig. 6. The ground truths are the output of the EBSD tool.

To quantitatively evaluate the algorithm accuracy an ImageJ library called MorphoLiJ is used [14, 15]. This library contains, among others, the plugin to evaluate the overlap agreement or error between labels. Given a source image \( S \) (algorithm output) and a target image \( T \) (ground truth), one can measure the values for all and individual regions [16]. Table 2 lists the measures with their definitions and the values obtained by comparing the algorithm output with the ground truths shown in Fig. 6. To be able to compare the results, the corresponding regions in the source and target image must have the same label, i.e. color. To this end, we created the label map for ground truth and then colored the regions in the source image accordingly. It needs to be mentioned that we performed the merging of the oversegmented regions in order to get a good estimation of how well the algorithm can detect the grain boundaries.

The main assumption for a good segmentation result is that the image to be analyzed is easily segmentable by the human eye. By taking a closer look at the original and ground truth images in Fig. 6, it can be noted that this is not the case. It would be very hard (if not impossible) even for a material scientist to accurately segment these images manually. The main problem is that some grain boundaries are...
very badly visible or not visible at all. Further, thin grains are usually not thicker than grain boundaries, so the algorithm cannot provide seeds for these regions. Taking all the above into account, the quantitative results shown in Table 2 are very good and indicate the potential of the proposed method.

4 Conclusions and Future Work

In this work, a clustering patch-based approach for grain detection in microstructure images is presented. Depending on the image quality and visibility of grain boundaries, the algorithm provides good and plausible results. Besides the patch size, no further user interaction is needed. The presented algorithm focuses on the distinction between grain and grain boundary patches based on different homogeneity criteria. To this end, GMM is used. The clustering output is used to create a seed image for the SRG algorithm. Such an approach makes it applicable for different types of microstructure images. Additionally, it can be used to automatically select seeds for the SRG algorithm in other application areas.

Since some grain boundaries in SEM images are not visible, an extension of the presented algorithm is required. It could be proven, that by tilting the sample in a SEM tool at different angles and using a specific detector, more information can be obtained. More precisely, different grains can be detected at different tilting angles, and thus images in which neighboring grains have different gray intensities can be obtained from the same material site. By combining this information, it should be possible to obtain an image in which all grain boundaries are present. The aim here is to reduce the number of necessary images to a minimum and to develop an image fusion method to combine all relevant information into one image. Finally, the presented segmentation algorithm will be applied and (if needed) adjusted for the fused image.

Abbreviations

EBSD: Electron Backscatter Diffraction; SEM: Scanning Electron Microscopy; FIB: Focused Ion Beam; SRG: Seeded Region Growing; NL-means: Non-local means; GMM: Gaussian Mixture Model; GLCM: Gray-level co-occurrence matrix; EM: Expectation maximization

Acknowledgements

The authors would like to thank Olivia Pfiefer and Barbara Pedretscher for valuable discussions on the topic.

Authors’ contributions

Both authors took part in the discussion of the work described in this paper. Both authors read and approved the final manuscript.

Funding

This work was funded by the Austrian Research Promotion Agency (FFG, Project No. 881110).
Availability of data and materials
All data generated or analysed during this study are included in this published article.

Competing interests
The authors declare that they have no competing interests.

Author details
1 KAI Kompetenzzentrum Automobil- u. Industrieelektronik GmbH, Europastraße 8, 9524 Villach, Austria.
2 Alpen-Adria-University, Department of Statistics, Universitätsstraße 65-67, 9020 Klagenfurt am Wörthersee, Austria.

References
1. Wang, L., Li, M., Almer, J., Bieler, T., Barabash, R.: Microstructural characterization of polycrystalline materials by synchrotron X-rays. Frontiers of Materials Science 7 (2013). doi:10.1007/s11706-013-0201-0
2. 4 - Strengthening of metal alloys. In: Mournitz, A.P. (ed.) Introduction to Aerospace Materials, pp. 57–90. Woodhead Publishing (2012). doi:10.1533/9780857095152.57. http://www.sciencedirect.com/science/article/pii/B9781855739468500042
3. Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(6), 641–647 (1994). doi:10.1109/34.295913
4. Gómez, O., González, J.A., Morales, E.F.: Image Segmentation Using Automatic Seeded Region Growing and Instance-Based Learning. In: Rueda, L., Mery, D., Kittler, J. (eds.) Progress in Pattern Recognition, Image Analysis and Applications, pp. 192–201. Springer, Berlin, Heidelberg (2007)
5. Fan, J., Zeng, G., Body, M., Hacid, M.-S.: Seeded region growing: an extensive and comparative study. Pattern Recognition Letters 26(8), 1139–1156 (2005). doi:10.1016/j.patrec.2004.10.010
6. Burger, W., Burge, M.J.: Digital Image Processing: An Algorithmic Introduction Using Java, 2nd edn. Springer-Verlag, London (2016). doi:10.1007/978-1-4471-4471-6-684-9
7. Buades, A., Coll, B., Morel, J.-M.: Non-Local Means Denoising. Image Processing On Line 1, 208–212 (2011). doi:10.5201/ipol.2011.bcm_nlm
8. Wagner, T., Behnel, P.: ij-nl-means: Non local means 1.4.6 (2016). doi:10.5281/zenodo.47468. https://doi.org/10.5281/zenodo.47468
9. Darbon, J., Cunha, A., Chan, T.F., Osher, S., Jensen, G.J.: Fast nonlocal filtering applied to electron cryomicroscopy. In: Proc. 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1331–1334 (2008). doi:10.1109/ISBI.2008.4541250
10. Haralick, R., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE Trans Syst Man Cybern SMC-3, 610–621 (1973)
11. Bishop, C.: Pattern Recognition and Machine Learning. Springer-Verlag, New York (2006)
12. Schneider, C.A., Rasband, W.S., Eliceiri, K.W.: NIH image to ImageJ: 25 years of image analysis. Nature Methods 9, 671–675 (2012). doi:10.1038/nmeth.2089
13. Schindelin, J., Rueden, C.T., Hiner, M.C., Eliceiri, K.W.: The ImageJ ecosystem: An open platform for biomedical image analysis. Molecular reproduction and development 82, 518–529 (2015). doi:10.1002/mrd.22489
14. Arganda-Carreras, I., Legland, D., Rueden, C., Mikushin, D., Eglinger, J., Burri, O., Schindelin, J., Heflich, S., Fiedler, C.C.: ijpb/MorphoLibJ: MorphoLibJ 1.4.2.1. doi:10.5281/zenodo.3826337. https://doi.org/10.5281/zenodo.3826337
15. Legland, D., Arganda-Carreras, I., Andrey, P.: MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. Bioinformatics 32(22), 3532–3534 (2016). doi:10.1093/bioinformatics/btw413. https://academic.oup.com/bioinformatics/article-pdf/32/22/3532/16920257/btw413.pdf
16. Tustison, N., Gee, J.: Introducing Dice, Jaccard, and Other Label Overlap Measures To ITK. The Insight Journal (2009)

Figures

Fig. 1 The main steps of the proposed workflow.

Fig. 2 The proposed workflow in details.

Fig. 3 The effect of different denoising methods.

Fig. 4 Results for FIB images.
Fig. 5 The effect of denoising methods on clustering output.

Fig. 6 Results for SEM images.