Machine Learning: Science and Technology

Improving the segmentation of scanning probe microscope images using convolutional neural networks

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
4 September 2020

REVISED
6 October 2020

ACCEPTED FOR PUBLICATION
5 November 2020

PUBLISHED
24 December 2020

PAPER

Improving the segmentation of scanning probe microscope images using convolutional neural networks

Steff Farley[1], Jo E A Hodgkinson[2], Oliver M Gordon[2], Joanna Turner[3], Andrea Soltoggio[1], Philip J Moriarty[2] and Eugenie Hunsicker[1]

[1] School of Science, Loughborough University, Epinal Way, Loughborough LE11 3TU, United Kingdom
[2] School of Physics & Astronomy, The University of Nottingham, University Park, Nottingham NG7 2RD, United Kingdom
[3] School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Epinal Way, Loughborough LE11 3TU, United Kingdom

E-mail: r.m.farley@lboro.ac.uk

Keywords: atomic force microscopy, dewetting, nanoparticle, machine learning, image segmentation, U-net

Supplementary material for this article is available online

Abstract

A wide range of techniques can be considered for segmentation of images of nanostructured surfaces. Manually segmenting these images is time-consuming and results in a user-dependent segmentation bias, while there is currently no consensus on the best automated segmentation methods for particular techniques, image classes, and samples. Any image segmentation approach must minimise the noise in the images to ensure accurate and meaningful statistical analysis can be carried out. Here we develop protocols for the segmentation of images of 2D assemblies of gold nanoparticles formed on silicon surfaces via deposition from an organic solvent. The evaporation of the solvent drives far-from-equilibrium self-organisation of the particles, producing a wide variety of nano- and micro-structured patterns. We show that a segmentation strategy using the U-Net convolutional neural network has some benefits over traditional automated approaches and has particular potential in the processing of images of nanostructured systems.

1. Introduction

Image segmentation is a common step in a workflow to carry out statistical image analysis. There are several techniques that can be employed for image segmentation and deciding upon one can be an important research question that can result in different analysis and conclusions. There is some existing knowledge on the strengths and weakness of various approaches but it is not always theoretically clear which techniques are optimum for particular patterns or noise [1]. For large datasets, it is also time consuming to choose (and optimise) a method for each image, and so a generalised automated approach is desirable.

In this paper we focus on a broad set of atomic force microscope (AFM) images of 2D nanoparticle assemblies formed on silicon substrates via deposition from a volatile organic solvent. The solvent dewetting process produces a wide variety of nanostructured patterns [2] spanning cellular networks, labyrinthine networks (bearing a striking similarity to those formed in polymer thin films), ensembles of islands, rings, and branched ‘tree-like’ structures, although this is not an exhaustive list. Analysis of the various patterns formed via evaporation of the solvent provides key insights into the dewetting process and enables the development of new protocols for the control of matter at the nanometre scale level [3–5].

AFM images of nanostructures present a further challenge for image segmentation. The scale of these structures require imaging techniques that are particularly susceptible to surface defects, contamination, and artefacts. When surfaces are imaged using a scanning probe, tip changes can result in loss of resolution, multiple-tip artefacts, discontinuities, and ‘smearing’. In addition, the properties of the piezoelectric actuators, and the design of the microscope itself, often lead to additional image distortions such as non-linear creep and thermal drift [6]. This is particularly problematic when comparing images from

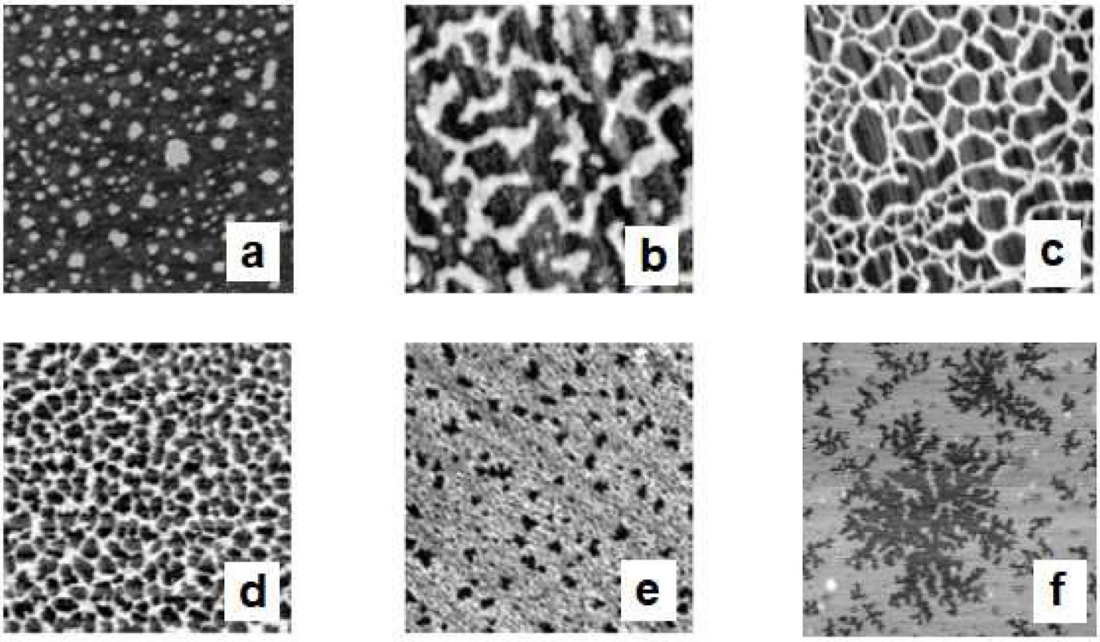

Figure 1. Characteristic examples of (a) islands, (b) worm-like domains, (c) cellular networks, (d) labyrinthine, (e) porous, and (f) fingering regimes identified in literature for AFM nanoparticle experiments.

experimental data to that of simulations produced by computational models which are free of the types of noise and distortion that plague experimental imaging.

A key issue, however, is that the majority of analysis/classification for nanostructured systems is currently carried out 'by hand' due to deficiencies in automated image processing when it comes to handling precisely the type of artefacts that are so prevalent in scanning probe data. Very often, the first step in the analysis of an AFM image of a nanostructured (or micro-structured) sample is segmentation in order to locate features. Here we consider traditional automated segmentation approaches and a neural network approach to segmentation and evaluate their ability to handle the types of artefacts found in this data. We aim to provide recommendations for the segmentation process from this evaluation.

2. Dataset and dataset curation

2.1. Nanostructured patterns

The dataset used contains (AFM) images of the depositions of gold nanoparticles that have been placed in a solvent and dried on a substrate. These tiff file images are 512 × 512 pixels in grey-scale format, with pixel intensity values between 0 (black) and 255 (white). The parameter space associated with the pattern generation process is very wide and encompasses variables such as the type of solvent, the concentration of nanoparticles, the length of the thiol chain surrounding the Au particle core, the relative humidity, the drying method (e.g. drop or spin cast), and the amount of excess thiol in the nanoparticle solution. See [7–10], for a description of the experimental protocols.

The solvent dewetting process leads to a number of distinct pattern types that have been described in the literature [2] as islands, worm-like, cellular, labyrinthine, pores, and fingering patterns. Examples of these regimes are shown in figure 1. Some regimes are qualitatively similar to each other in terms of visual interpretation. Worm-like features could be considered as stretched out islands, and cellular images are similar to connected worm-like features. Labyrinthine structures are similar to cellular image but have less of a tiling pattern. Pores may be seen as more sparsely populated and less ramified labyrinthine structures. Finally, fingering features could be seen as a variation of pores. There are some images that lie on the border between two of these regimes in this phase diagram. We also see images with two or more regimes represented, as well as multi-scale examples of regimes.

We sometimes also observe two or more layers in these images, where an additional layer of nanoparticles has formed, and the pixel intensity of nanoparticles on different layers can vary. An example is shown in figure 2. This occurs when the nanoparticle solution has a relatively high concentration [11]. In this paper we focus on binarisation approaches for single-layer images as multi-layer images require multi-level segmentation approaches which are not evaluated in this paper.

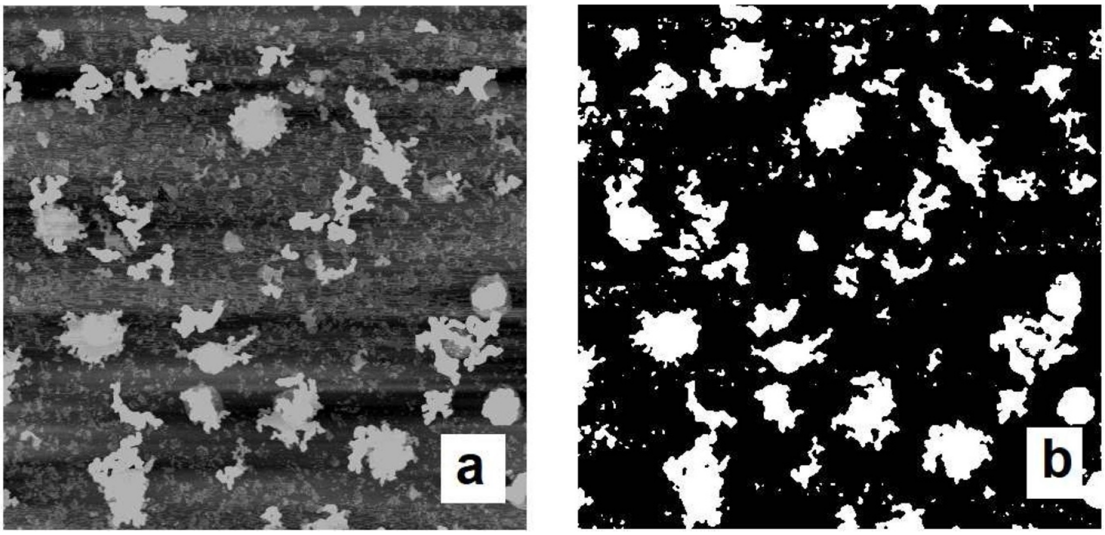

Figure 2. Example of a multi-layer image and an attempted segmentation of this image using a conventional binary segmentation method.

Developing a generalised, accurate and repeatable segmentation method for these images is an important task as small changes in threshold can lead to different segmented images and subsequently substantially different extracted statistics. To demonstrate this, we took one characteristic example of a fingering regime and one characteristic example of a worm-like regime and segmented at four different thresholds for each image. These images were chosen because fingering and worm-like images are qualitatively different. The original images are shown in figure 3. After choosing a threshold value, all pixels with an intensity below this value are set to black and all pixels above the threshold value are set to white, converting the grey-scale image to a binary image. By taking four different thresholds we can observe the variability of common statistics that can be extracted from these images and used in metrics. We used the 2D Minkowski numbers—perimeter, area, and Euler characteristic—as statistical measures. These have been used previously to analyse the distribution of features in these images [12, 13].

The four different thresholds were chosen to cover a range of threshold values such that they spanned the range of thresholds manually chosen by seven individuals, provided in table 1 in the supplementary information (available online at stacks.iop.org/MLST/2/015015/mmedia). These individuals were asked to set a threshold that they believed gave the best binarisation for these images. From the images in figure 3 we can see that the visual difference between these images is not substantial but normalised Minkowski numbers extracted from each of these images showed a substantial variation across the thresholds. See table 3 in the supplementary information for the full results of this experimentation. The worm-like image was particularly sensitive to the choice of threshold. This shows us that repeatability, along with accuracy, is an important quality for any segmentation approach.

2.2. Pre-processing

Scan data returned by scanning probe microscopy can be read into software as grey-scale 2D images. Pixel coordinates represent physical tip position on the scan surface, while pixel intensity values within a single value channel provide representative heights of surface features, providing both effective visual and topographical information. Popular image processing software, such as ImageJ [14] and Gwyddion [15], focuses on loading individual images, where processing methods and their parameters are controlled by the user based on the appearance of the resulting images. This can lead to biases towards visual information as opposed to complete topographical accuracy. A processed image tolerable by eye for the purpose of presentation often contains bias towards removal of local defects and enhancement of desired surface features, resulting in an image that may still have large quantities of noise that will reduce the quality of its future segmentation. Alternatively, an automated pre-processing regime is less time-consuming than manual processing within a popular image processing software, and promotes consistency in image statistics. Pre-processing software using a combination of python and R language code was adopted.

Raw data obtained directly during routine AFM experimentation will contain artefacts produced by the scanning probe. Line-by-line acquisition results in ubiquitous scan line artefacts, leading to incongruity particularly visible in multi-layer images, while raw output data is not systematically leveled which significantly affects the quality of binarisation. The proposed pre-processing software attempts to remove or

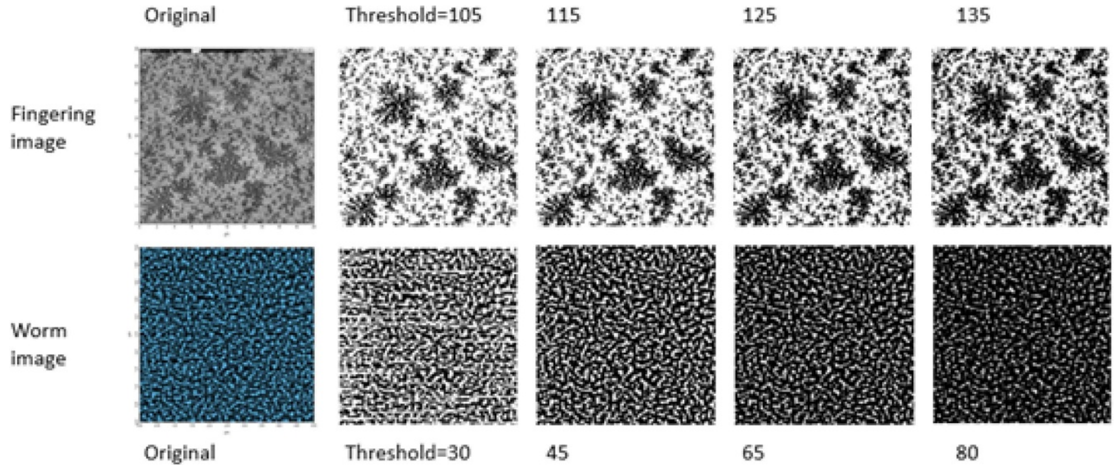

Figure 3. Images taken at a series of thresholds that cover the range of threshold values subjectively chosen by seven individuals: 105, 115, 125, and 135 for a fingering image and 30, 45, 65, and 80 for a worm-like image.

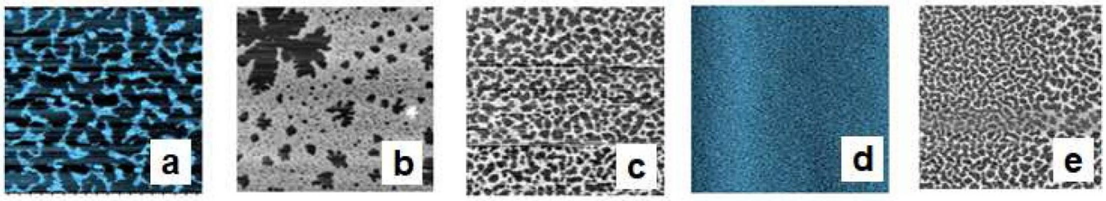

Figure 4. Examples of (a) streaking, characterised by darker regions in the image being brightened in horizontal streaks, (b) background contrast, occurring when the pixel intensity varies noticeably in large lighter regions, (c) striping, characterised by solid, thin horizontal lines, (d) banding, a difference in brightness in a vertical band within the image, and (e) blurring noise, typically seen as a blurring effect contained in horizontal band.

reduce scanning artefacts using two algorithms to return a data-leveled image with spatially-continuous layers. A corrective line-by-line alignment algorithm is deployed along the same axis as line-by-line scans. Pixel values in each row are shifted to have the median difference between pixels in neighbouring rows in the nearest row equal zero. Median of differences row alignment was found to be the most effective technique for creating continuous features but does not sufficiently level the image. An orthogonal polynomial detrending algorithm with three degrees of freedom returned sufficiently leveled images; the low degree of freedom results in a general image background subtraction without data loss due to equations fitting to data instead of noise. Subtle or local fluctuations in images identified by higher degrees of freedom fits heavily diminished during binarisation.

The contrast of these images is then normalised such that the minimum pixel value becomes 0 and the maximum value becomes 255, using a combination of shifts and scaling operations [16]. In some images, the minimum and maximum pixel intensities are already at these values. Therefore pixels outside a certain distance from the mean are identified and we then apply contrast normalisation excluding these pixels, which are then set to 0 or 255 by default depending on whether they were smaller or larger than the mean. We consider using 3, 2, or 1 standard deviations from the mean and evaluate which option provides the best segmentation results.

2.3. Noise

The common types of noise we encountered can be described as banding, streaking, background contrast, stripes, and blurring. Examples of each of these are given in figure 4. These types of noise have been observed and discussed in previous literature on scanning probe microscope (SPM) images [17]. We can also see the relevance of such noise by the fact that packages with image processing functions have been developed, such as Gwyddion [15], to deal with these types of noise, albeit not in an automated manner.

The most problematic noise types were banding, background contrast and streaking, as we observed that these had the greatest impact on statistics extracted from the images when they were artificially added to relatively clean examples. We demonstrated this by selecting 28 images that were accurately segmented using two common methods: Otsu's threshold [18] and a local mean threshold [19]. Otsu's threshold chooses a threshold value such that the variation within the two groups (pixels set to black and pixels set to white) is minimised, or rather, the inter-class variance is maximised. Local mean thresholding uses an adaptive mean in a moving window within the image, such that each pixel has a different threshold value determined by the mean pixel intensity in the window surrounding it.

We then artificially added the five most common noise types to these images. Using the same segmentation methods, we binarised these new noise augmented images and again extracted Minkowski numbers, observing how these these statistics changed in noise-augmented images from the originals. For the Euler characteristic, we measured the average difference between noise augmented images and the originals. For area and perimeter we measured the average difference relative to the mean. The largest average differences were seen for background contrast, banding, and streaking, suggesting that these noise types have the greatest impact on accurate segmentation. See table 1 in the supplementary information for the full results of this experimentation.

2.4. Dataset curation

To evaluate segmentation approaches, we need a dataset that is representative of the structures and noise that are observed in these images. For training a network for semantic segmentation, we need several examples of each structure and noise type to construct a training dataset that produces a network with sufficient accuracy. For each training image, we produce a corresponding segmented image from either manual segmentation or from using an automated segmentation method with sufficiently accurate results by visual inspection. We also need a representative testing dataset to evaluate the segmentation approaches.

The full dataset consisted originally of 2625 images, taken at various scales. Images from the six regimes identified above from the literature were first identified and included in the study set. From the remaining images, other regimes that appeared less frequently in the dataset were identified, including rings, trees, streaks, and cracks. All images from those regimes were added to the new dataset. Images with two or more regimes present were also included, as well as images with multiple layers. For those with multiple layers, we segmented the most prominent layer. Those with excessive noise or no clear regime were not included in either the training or testing set. An example of an image with excessive noise is given in figure 1 in the supplementary information. This left 728 remaining images in the dataset. For each of these images, the regime and noise were manually labelled by visual observations and recorded.

This set was separated into training and testing sets by choosing approximately 75% of images from each domain to be training data, stratifying by noise type. This was achieved by assigning a random number between 0 and 1 to each image and selecting approximately the first 25% of images with the greatest number from each domain and noise type for the testing set. When there was only a single case of a particular noise type for a particular domain, that image was included in the testing set if the random number associated with it was greater than 0.5. We therefore produced an original training dataset of size 428 and a testing dataset of size 300. Each image has the following fields associated: regime, noise, scale and multi-layer indicator flag. A breakdown of these datasets by regime and noise type can be found in table 4 in the supplementary information.

2.4.1. Augmentation

To ensure that the training dataset was representative of all noise types for all regimes, we augmented a subset of 80 originally clean images with artificial noise. We consider three noise augmentation processes. For Noise Augmentation Process 1, the types of noise that we introduced were multiple stripes, banding, streaking, background contrast, inversion, blurring, drift, and a combination of banding and stripes. This produced 560 additional images to be included in the training dataset. Examples of an original image and added noise are shown in figure 5. Streaking was created by applying a mask to black pixels the image. The inverse of the mask was also applied to white pixels in images to create an augmented image with varying background contrast. This mask could be augmented with flips and scales to vary the streaking between images. The mask, shown in figure 5, was taken from a streaking image from the training dataset. As we already have the manual segmentations of the clean images that we were artificially adding noise to, we were able to use the same segmented images for training networks for both the clean and noisy examples.

For Noise Augmentation Process 2, we introduced the same noise as in Process 1 but replaced the streaking mask with a high frequency horizontal banding, which also resulted in the same number of additional images for training (560). For Noise Augmentation Process 3, we removed uncommon noise types and only augmented with stripes, banding, streaking with mask, background contrast, and a combination of banding and stripes. Process 3 provides us with 400 augmented images.

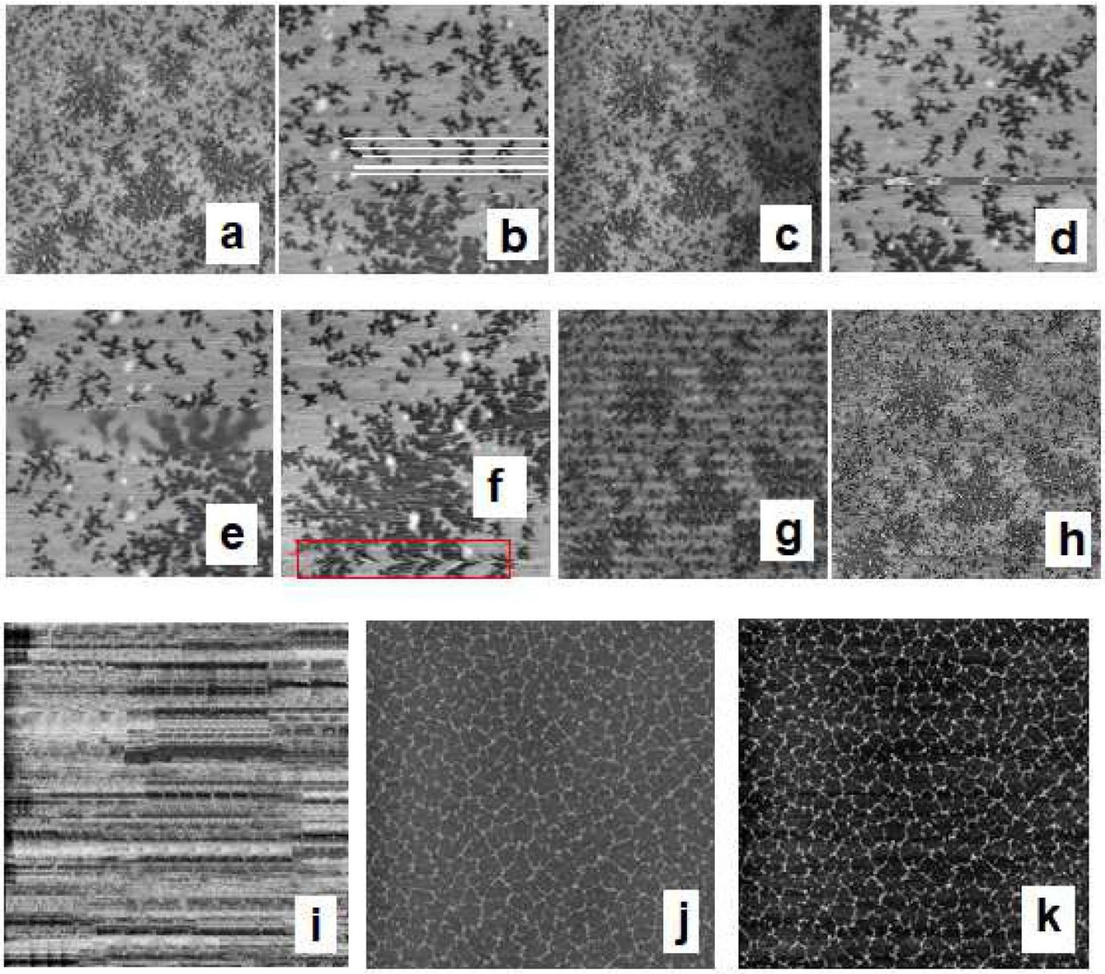

Figure 5. Examples from artificial noise augmented process 2 to clean images. (a) Original image, (b) stripes, (c) vertical banding, (d) inversion, (e) blurring, (f) drift (in red box), (g) streaking with high frequency horizontal banding, (h) varying background contrast. Image (i) shows a mask and (k) demonstrates this mask applied to image (j) to imitate streaking.

Table 1. The pre-processing, binarisation and post-processing approaches that were reviewed.

Pre-processing	Binarisation	Post-processing
K-means, $K > 2$	Global mean threshold	Despeckling
Mean shift	Local mean threshold	Autoencoder
Gaussian filter	Global Otsu's threshold	
Histogram equalisation	U-Net	

3. Methodology

3.1. Segmentation methods

In this work we have reviewed several binarisation approaches in combination with pre-processing operations. A complete list of pre-processing, binarisation and post-processing approaches considered is given in table 1. Each method was applied to the images in the testing dataset.

K-means clustering with $K > 2$ and mean shift clustering both carry out multi-class image segmentation but can be used as a pre-processing operation to deal with some noise in images. We used code written by A. Asvadi [20] to implement these techniques. A Gaussian filter [21] is another pre-processing approach commonly applied to noisy images, particularly those with small enclosed regions of black or white pixels, called 'salt and pepper' noise. This results in a slightly blurred image, dealing with noise but also reducing the quality of the image. An image from the dataset and the same image after k-means clustering with $K = 8$, mean shift clustering, and a Gaussian filter are shown in figure 6. The last pre-processing approach we considered was histogram equalisation. This is a common technique to normalise the contrast in the images

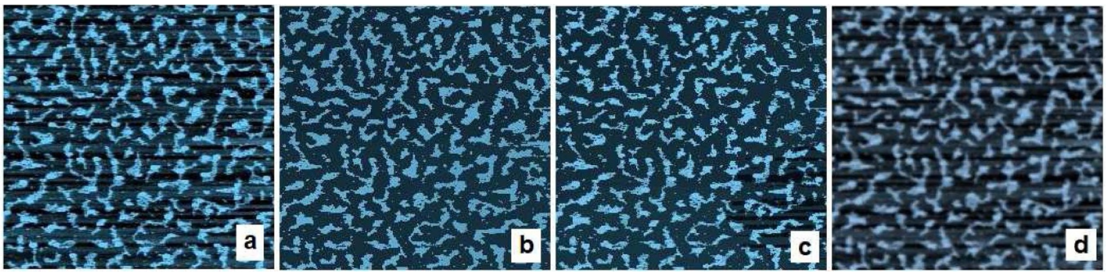

Figure 6. (a) Original image, (b) image with K-means segmentation, $K = 8$, (c) image with mean shift segmentation, (d) image with Gaussian filter.

Table 2. U-Net model trained with different settings.

Name	Noise augmentation	Contrast normalisation
Unet1	Noise augmentation type 1	2 standard deviations
Unet2	Noise augmentation type 3	2 standard deviations

before binarisation techniques are applied by attempting the create a uniform distribution of pixel intensities in the image [22].

For binarisation approaches, in addition to Otsu's threshold and local mean threshold introduced in section 2.3, we also consider a simple thresholding approach that uses the mean of all pixel intensities in the image as the global threshold value. Any pixels above this value are set to white and any below this value are set to black. Otsu's threshold considers more information about the pixel intensity distribution and as such is expected to perform better than the global mean threshold. However, it assumes that the distribution of recorded scan heights is bi-modal.

We also consider a machine learning approach to segmentation. U-Net is a convolutional neural network architecture designed for segmenting images, where the task is to predict a classification for each pixel [23]. This architecture has been shown to achieve state-of-the-art performance for biomedical image segmentation [24]. Its unique U-shaped architecture, made up of a contracting path and an expanding path, facilitates the use of contextual information to inform pixel predictions, and precise localisation. The task of nanoparticle deposition segmentation has similar requirements to biomedical segmentation, it was therefore hypothesised that the U-Net architecture could be well suited for this problem. Precise localisation is required to infer accurate and meaningful statistics. Contextual information in the image can provide clues about the type of regime and therefore may assist with the predictions of class for each pixel. The dataset we curated was used to train this network. The U-Net training strategy can address the need to learn invariance to challenging noise types, such as banding, contrast and streaking, with only few labelled images for training (428).

We trained the network on the training dataset that we curated and then applied the trained network to the testing dataset in the same way as the other segmentation methods we are evaluating. We consider this network with and without the noise augmented images included in the training dataset. We also considered the different noise augmentation options, different pixel histogram truncation before contrast normalisation, and the order in which we performed noise augmentation and contrast normalisation. We evaluated several models but in this paper we focus on the two best performing configurations, that we named Unet1 and Unet2. See table 5 in the supplementary information for details on the settings used for training these U-Net models.

After image segmentation, we also consider two post-processing techniques. Despeckling is a post-processing technique that replaces each pixel with the median value within its 3 × 3 pixel neighbourhood [14] and is effective at removing salt and pepper noise. The second post-processing method we consider is a denoising autoencoder. This has been successfully employed to improve classification of AFM images of nanoparticle depositions [25]. Input images are encoded and then decoded by an autoencoder, before the reproduced, original image is outputted. The autoencoder developed and described in Gordon *et al* [25]. was trained on simulated images with artificial salt and pepper noise and taught to output the original image without this noise.

3.2. Evaluation methods

We evaluated these methods through a combination of qualitative and quantitative analyses of the segmented testing images they produced. We first evaluated the methods visually through manual observations of the

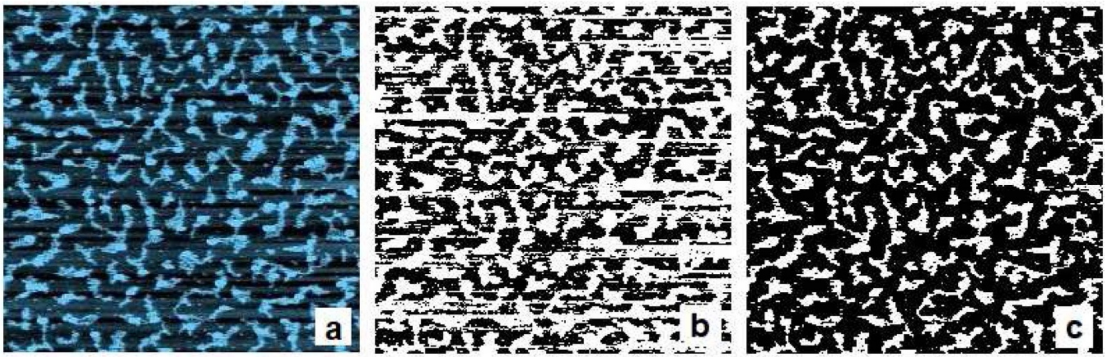

Figure 7. An image segmented by local mean where (b) histogram equalisation was applied prior to segmentation and (c) where histogram equalisation was not applied.

segmentations, providing subjective comments on the performance of these methods for different noise types in the dataset. We gave subjective interpretations of which segmentation methods performed best for each testing image by visually determining how well image feature details are maintained and noise artefacts are minimised in the segmented images. For each segmentation method, we also recorded the number of testing images where substantial noise artefacts remained and the method failed to produce an adequate segmentation.

The performance of each segmentation method was also compared quantitatively by observing the proportion of pixels that changed in segmentations when artificial noise was added for 16 images. We selected 16 images, for which each segmentation method was capable of segmenting well, and artificially added banding, blurring, and striping noise to these examples. Using the same segmentation methods, we binarised these new noise augmented images and determined how many pixels changed (from black to white or from white to black) in the segmented noise augmented images from the segmented original clean images. We determine the proportion of pixels that changed in the 16 images for each artificial noise type and each segmentation method.

The features in these 16 images were relatively small, as we chose images for which the segmentation methods perform well and local mean struggles to correctly segment larger features even in the absence of noise. Streaking and varying background contrast have less of an impact on segmentation when features are small so to evaluate the methods on these noise types, we split each image into four quarters and rescaled each sub-image to 512 × 512 pixels to increase the size of the features. This created four images from each original image, providing a total of 64 images for this analysis. We artificially added varying background contrast and streaking to these images and segmented them using the four methods we are evaluating. We again determined how many pixels changed from these images to the same sub-images (rescaled) from the original segmentations where no artificial noise is present.

4. Results

4.1. Qualitative analysis

We can make some simple initial observations through evaluating the identified segmentation approaches visually on the 300 single-layer testing images that were representative of all regimes and noise types. The first result we observed was that histogram equalisation in combination with any other pre-processing method and any binarisation approached performed poorly, created more noise than the same approaches without histogram equalisation and often produced completely unusable binarised images. We also found that using k-means or mean shift resulted in information lost before image segmentation without dealing with noise. Applying a Gaussian filter also rarely improved segmentation and typically lead to coarse features. Using no additional pre-processing to alignment, detrending, and contrast normalisation, typically produced the best results.

For binarisation methods, as there were different types of noise that appeared in different images, no single option appeared to be able to deal with all of them optimally and some options appeared to perform better on certain types of noise and other options were more suited to different types of noise. For example, figure 8 shows how using a local mean threshold deals well with banding in the cellular image where Otsu's threshold still contains substantial banding, however, the local mean approach contains streaks in the worm-like image, which do not appear when using Otsu's threshold. This is a trend we observed throughout

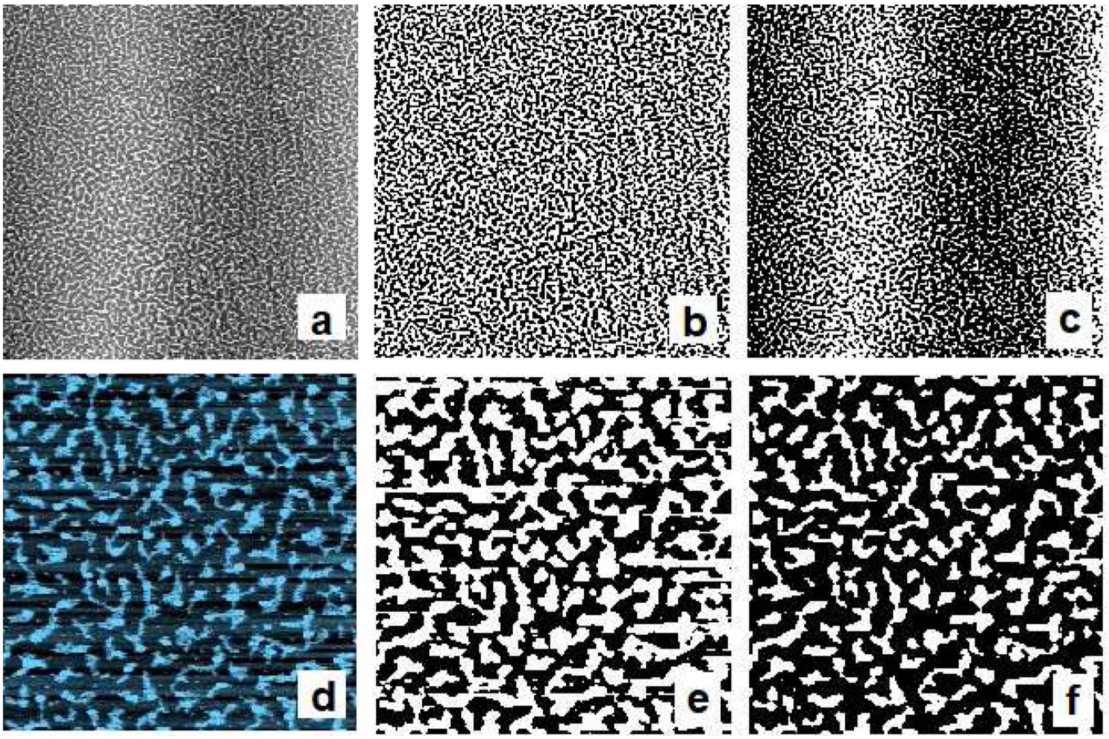

Figure 8. The top row shows (a) an image with banding segmented with (b) local mean and (c) Otsu's threshold. The bottom row shows (d) an image with streaking segmented with (e) local mean and (f) Otsu's threshold.

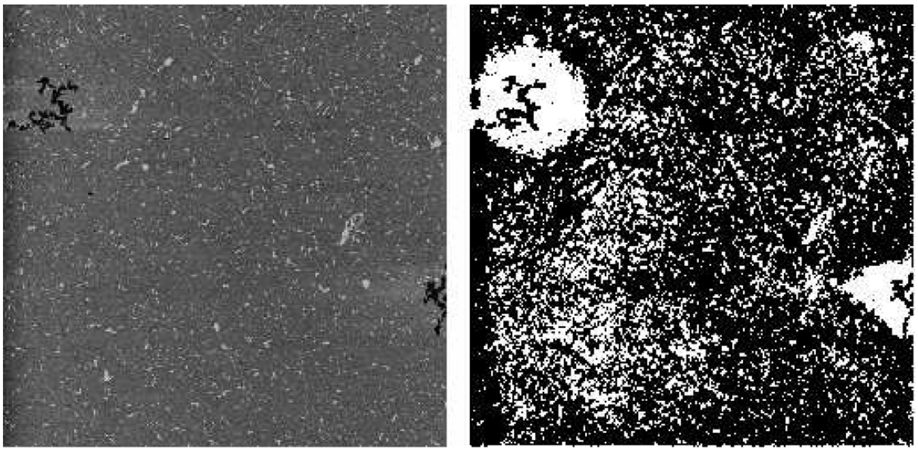

Figure 9. Example of a multi-layer image segmented using a U-Net model.

the testing dataset. Where Otsu's threshold performed best for streaking, it was also usually improved by applying despeckling too.

We then evaluated the performance of the U-Net networks for segmentation. We note that the U-Net models trained on two class segmentations fail with multi-layer images, as seen in the example in figure 9. This is because the model has been optimised for single-layer images, which have been the focus of this paper, and the U-net model can only be evaluated on multi-layer images when further work has been done to train the U-net model on these images. The only approach we evaluated that deals with multi-layer images well is Otsu's threshold. We therefore only evaluated the binarisation approaches on the remaining 107 single-layer testing images.

We also notice that Unet1 and Unet2, which were trained on more complex noise augmentation and contrast normalisation, were significantly better than a basic U-Net model trained on no noise augmented data. We compared the segmentations of the 107 single-layer testing images for Unet1 and Unet2 and observed that Unet1 appeared to perform better on streaking and contrast noise, whereas Unet2 appeared to deal with banding noise better, but both U-Net models handled all of these different noise types reasonably

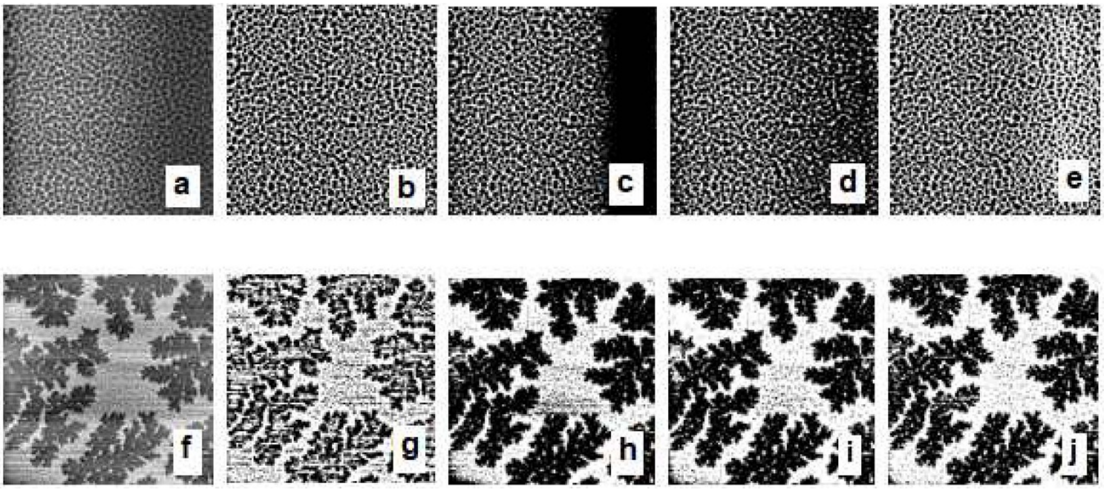

Figure 10. (a) Original image with banding, (b) image segmented by local mean, (c) by Otsu's threshold, (d) by Unet1, (e) by Unet2, (f) original image with streaking and varying background contrast, (g) image segmented by local mean, (h) by Otsu's threshold, (i) by Unet1, (j) by Unet2.

Table 3. Number of testing images for which each segmentation method gave the best segmentation, according to subjective manual observations, for different noise types. The number of images for which each segmentation method failed to produce an adequate segmentation is also recorded.

	Local mean		Otsu's threshold		Unet1		Unet2		
	Best	Failed	Best	Failed	Best	Failed	Best	Failed	Number of images
Background contrast, streaking and/or stripes only	30	27	43	21	42	21	30	19	65
Banding only or banding with stripes only	10	1	8	5	6	2	14	0	21
Blurring	5	2	7	2	7	2	7	1	11

well. For the 107 single-layer testing images, Unet1 performed best according to subjective manual observation in 62 images, Unet2 performed best in 60, Otsu's threshold with despeckling performed best in 65, and local mean with despeckling performed best in 56, where multiple approaches can perform equally well for the same image. The methods performed similarly in the total number of images they were able to give the best segmentation for but this changed for different noise types. Moreover, the U-Net models had a tendency to give reasonably good segmentations for images that they did not perform best for, and there were very few images where they substantially failed to give a good segmentation. This is contrary to the Otsu's threshold and local mean methods, that often failed completely for banding and streaking noise respectively. Table 3 shows how each method performs for different noise types.

In testing images that contained only streaking, contrast, or stripes, and no other noise types, Unet1 and Otsu's threshold appeared to perform best according to subjective visual evaluation of the images. In images that only contained banding or banding and stripes, Unet2 and local mean with despeckling both performed best. These results show that Unet1 appears to perform similarly to Otsu's threshold in images that lend themselves to the strengths of Otsu's threshold, while Unet1 also failed for fewer images than Otsu's threshold for images containing banding. While Unet1 is weaker at banding than local mean, it is able to deal with streaking noise much better. Unet2 performed substantially better than Otsu's threshold at banding and gave better results than local mean for streaking. This provides some evidence that it is possible to train a U-Net model for a more generalised approach to segmentation of these images. Additional experimentation with the training dataset and the settings of the network could improve on these results, finding a balance between the strengths of Unet1 and Unet2, and providing a more robust and generalised segmentation method.

The denoising autoencoder was also applied to segmentations from this dataset in Gordon *et al* [25] and compared to the same segmentations without the autoencoder to determine whether this resulted in improvements to the segmentations for purpose of image classification. They found that it had some success at denoising striping and streaking artefacts. As the autoencoder was trained to remove speckle noise, it was

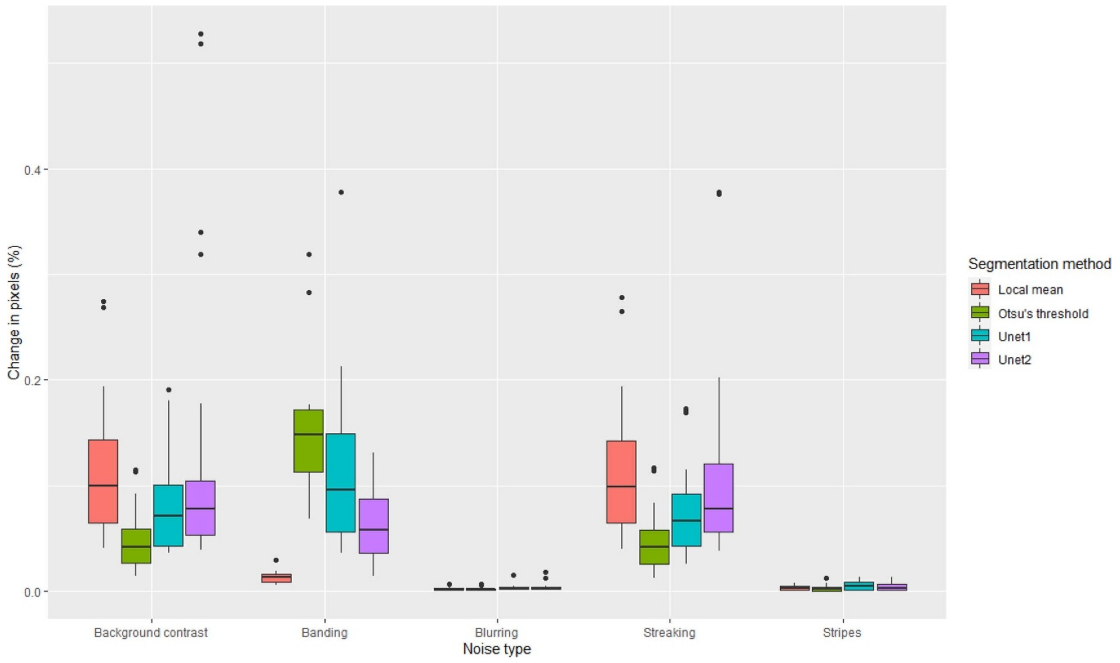

Figure 11. Proportion of pixels changed when different noise types are artificially added to images and segmented with different segmentation options.

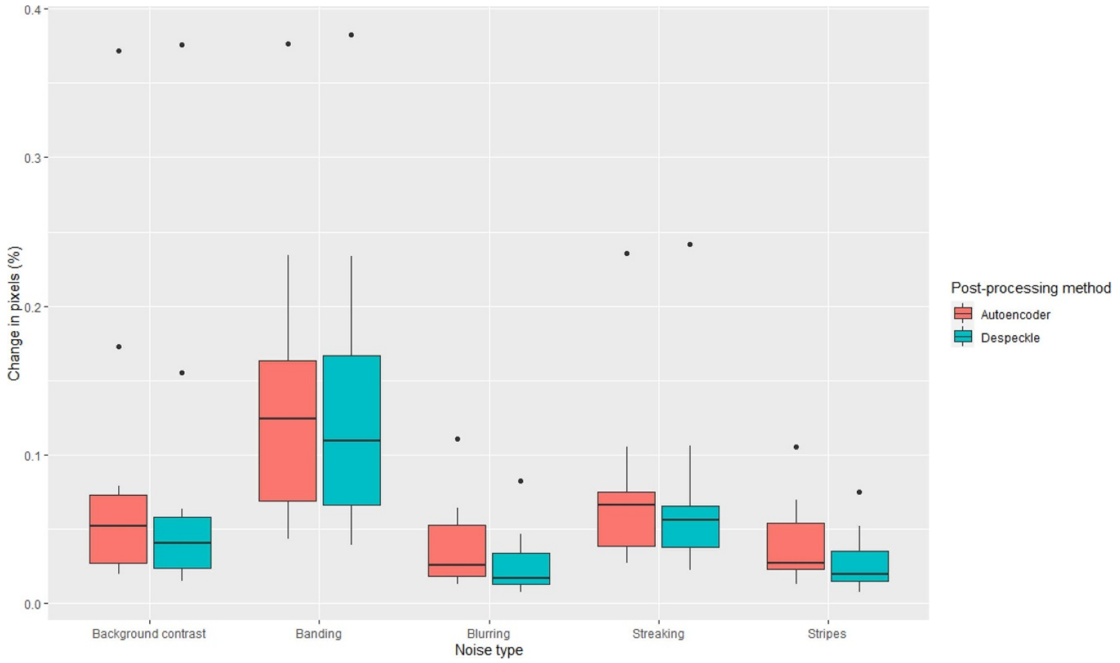

Figure 12. Proportion of pixels changed when noise is artificially added to images and segmented by Unet1 with different post-processing methods.

expected that it would perform well on the thin artefacts associated with striping and streaking noise. These results suggest that applying an autoencoder as part of post-processing improves the results of segmentation for methods that had relatively greater difficulty with streaking, such as the U-Net models.

4.2. Quantitative analysis

We also determined the number of pixels that were altered with each segmentation method when artificial noise was added to clean images. Results for artificial banding, blurring, and striping noise applied to 16 clean images and for background contrast and streaking noise, which were added to 64 rescaled sub-images, are given in figure 11. As expected, the U-Net models both showed improvement on handling banding noise as compared to the values from Otsu's threshold but neither were able to match the performance of local mean for this type of noise. We found that Unet2 particularly outperformed Otsu's threshold for banding noise and Unet1 handled both varying background contrast and streaking better than local mean. Unet1 was also more comparable to Otsu's threshold for background contrast and streaking noise types than Unet2. This analysis produced results that match and support the manual observations from the qualitative analysis.

We additionally measured the effect of applying post-processing to the Unet1 segmentation and compared the performance of despeckling and the denoising autoencoder as post-processing methods. We found that these methods performed similarly and, while both proved capable of removing artificial streaking and striping noise, both methods also removed features in the images that were not noise artefacts. These results are shown in figure 12. There is a trade-off as both methods remove noise and some features but, depending on the application and the goals of segmentation, this post-processing may be desirable. The autoencoder has shown to be favourable for improving classification rate [25], for example, but using the autoencoder or despeckling would not be desirable if the goal of segmentation is to study the fine structure of the material.

5. Conclusion

We have investigated a machine learning approach for binary segmentation of noise images and developed an approach that performs as well as traditional methods where they succeed and outperforms on images where each traditional method fails. The evaluation of approaches showed that local mean and Otsu's threshold behave very differently and perform best on different images, whereas the network we have developed incorporated the best of both of these approaches. The U-Net models we developed do not always necessarily provide the best segmentations but typically outperform either Otsu's threshold or local mean, depending on the type of noise present, as shown in table 3, figures 10 and 11. Unet2 failed for fewer images containing streaking or background contrast than local mean, and consistently outperformed Otsu's threshold and Unet1 for banding noise. Whilst Otsu's threshold outperforms Unet2 for streaking and background contrast noise, Unet1 showed some improvement over Unet2 for this noise type and was more comparable to Otsu's threshold where Otsu's threshold performs well, showing that, with more training, there is potential for a U-Net model that performs well for all noise types. As there was no single general method that worked best for all images, future research may also include a machine learning approach to recognise noise in images and automate the identification of the most suitable segmentation method.

Through some modification to the size of the training dataset, the addition of augmented noisy images, and changes to the contrast normalisation, we explored ways of improving the network. This showed that some choices in training dataset curation can significantly alter the performance of the network and further attention to this may lead to greater improvements. Further improvements may also be found through altering the settings and structure of the network. The results of this research show that this would be worthwhile endeavour.

While the U-Net network did not perform well for multi-layer images, a different training dataset with multiple labels to annotate different layers may be used to similarly develop a network that can handle the multi-classification problem of multiple layers. The results for single-layer images provides optimism for the use of a machine learning approach for multi-layer images.

Supporting information available

- A complete dataset of all pre-processed images from the training and testing datasets, the training labels, and the Unet1 model are openly available at https://doi.org10.17639/nott.7072.
- Code is available at https://github.com/steff-farley/AFM-image-segmentation.

Acknowledgment

The authors gratefully acknowledge Emmanuelle Pauliac-Vaujour who also produced the dataset of SPM images used in this work.

ORCID iDs

Steff Farley https://orcid.org/0000-0001-5335-0328
Oliver M Gordon https://orcid.org/0000-0001-8733-7500
Philip J Moriarty https://orcid.org/0000-0002-9926-9004

References

[1] Davies E R 2012 *Computer and Machine Vision* 4th edn (Amsterdam: Elsevier) pp 82–110 (available at: https://doi.org/10.1016/B978-0-12-386908-1.00004-5)

[2] Stannard A 2011 Dewetting-mediated pattern formation in nanoparticle assemblies *J. Phys. Condens. Matter* **23** 3001–20

[3] Vlasov Y A, Bo X, Sturm J C and Norris D J 2001 On-chip natural assembly of silicon photonic bandgap crystals *Nature* **414** 289–93

[4] Thiele U, Mertig M and Pompe W 1998 Dewetting of an evaporating thin liquid film: heterogeneous nucleation and surface instability *Phys. Rev. Lett.* **80** 2869

[5] Maillard M, Motte M, Ngo A T and Pileni M P 2000 Rings and hexagons made of nanocrystals: a Marangoni effect *J. Phys. Chem.* B **104** 11871–7

[6] Straton J C, Bilyeu T T, Moon B and Moeck P 2014 Double-tip effects on scanning tunneling microscopy imaging of 2D periodic objects: unambiguous detection and limits of their removal by crystallographic averaging in the spatial frequency domain *Cryst. Res. Technol.* **49** 663–80

[7] Pauliac-Vaujour E and Moriarty P 2007 Meniscus-mediated organization of colloidal nanoparticles *J. Phys. Chem.* C **111** 16255–60

[8] Pauliac-Vaujour E *et al* 2008 Fingering instabilities in dewetting nanofluids *Phys. Rev. Lett.* **100** 176102

[9] Martin C P *et al* 2007 Controlling pattern formation in nanoparticle assemblies via directed solvent dewetting *Phys. Rev. Lett.* **99** 116103

[10] Blunt M O *et al* 2007 Coerced mechanical coarsening of nanoparticle assemblies *Nat. Nanotechnol.* **2** 167–70

[11] Martin C P *et al* 2008 Chapter 1: Self-organised nanoparticle assemblies: a panoply of patterns *Stud. Multidiscip.* **5** 1–20

[12] Martin C P, Blunt M O, Moriarty P 2005 *Gold Nanoparticles Under the Microscope* Imaging & Microscopy (available at: www.imaging-git.com/science/image-processing/gold-nanoparticles-under-microscope)

[13] Stannard A *et al* 2008 Dual-scale pattern formation in nanoparticle assemblies *J. Phys. Chem.* C **112** 15195–203

[14] Schneider C A, Rasband W S and Eliceiri K W 2012 NIH Image to ImageJ: 25 years of image analysis *Nat. Methods* **9** 671–5

[15] Nečas D and Klapetek P 2012 Gwyddion: an open-source software for SPM data analysis *Cent. Eur. J. Phys.* **10** 181–8

[16] Bradski G 2000 The OpenCV Library Dr. Dobb's Journal of Software Tools (San Francisco, CA: UBM Technology Group) 25 120, 122–5 (https://www.drdobbs.com/open-source/the-opencv-library/184404319)

[17] Stirling J, Woolley R A J and Moriarty P 2013 Scanning probe image wizard: a toolbox for automated scanning probe microscopy data analysis *Rev. Sci. Instrum.* **84** 113701-1-6

[18] Otsu N 1979 A threshold selection method from gray-level histograms *IIEEE Trans. Syst. Man Cybern.* **9** 23–7

[19] Dey N *et al* 2014 Adaptive thresholding: a comparative study *2014 Int. Conf. on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)* (https://doi.org/10.1109/ICCICCT.2014.6993140)

[20] Asvadi A 2015 *K-means, mean-shift and normalized-cut segmentation* Code (available at: https://www.mathworks.com/matlabcentral/fileexchange/52698-k-means-mean-shift-and-normalized-cut-segmentation)

[21] Leach R K 2010 Chapter 8—surface topography characterization *Fundamental Principles of Engineering Nanometrology* (Amsterdam: Elsevier) pp 211–62

[22] McReynolds T and Blythe D 2005 *Advanced Graphics Programming Using OpenGL* (Amsterdam: Elsevier Inc) pp 211–45

[23] Zhixuhao U 2017 [GitHub repository] (available at: https://github.com/zhixuhao/unet)

[24] Ronneberger O, Fischer P and Brox T 2015 U-Net: convolutional networks for biomedical image segmentation *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015 MICCAI* vol 9351 eds N Navab, J Hornegger, W Wells and A Frangi (*Lecture Notes in Computer Science*) (Cham: Springer) (available at: https://doi.org/10.1007/978-3-319-24574-4_28)

[25] Gordon O*et al* 2020 Automated searching and identification of self-organised nanostructures *Nano Lett.* **20** 7688–93