Image processing towards the automated identification of nanoparticles in SEM images

Nicolene Botha
Modelling and Digital Science (MDS)
Council for Scientific and Industrial Research (CSIR)
Stellenbosch, South Africa
NBotha@csir.co.za

Gert Wessels
Modelling and Digital Science (MDS)
Council for Scientific and Industrial Research (CSIR)
Stellenbosch, South Africa
GWessels@csir.co.za

Natasha Botha
Modelling and Digital Science (MDS)
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
NBotha1@csir.co.za

Beatrice van Eden
Modelling and Digital Science (MDS)
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
BvEden@csir.co.za

***Abstract*—SEM images are crucial in the characterisation of material properties. These images can be very hard to interpret without any prior knowledge of the material. This paper discusses a pre-processing method for assisting convolutional Neural Networks in identifying the presence of nanoparticles in composite SEM images. The pre-processing method is developed using a synthetic SEM image.**

Keywords—image processing, scanning electron microscopy, composite materials, LULU smoothing

I. INTRODUCTION

Nanomaterials are versatile, when designing new materials for specific applications, due to the unique properties that can be obtained by mixing different base (e.g. polymer) and filler (e.g. nanoparticles and nanofibers) materials [1-2]. Therefore, understanding the behaviour, interaction and correlation of the resulting composite material properties with the size and distribution of the nanoparticles is of interest to material scientists, engineers and designers [1-3]. The general practice to obtain size distribution of nanoparticles is by manually measuring these variables from scanning electron microscopy (SEM) and transmission electron microscopy (TEM) images. SEM provides information on the material composition and surface while TEM provides further insights into the morphology and crystal structure [4].

Analysing composite nanomaterial images are difficult as there are two different materials that should be distinguished before any useful information can be extracted. An example of such a complex material is shown in Fig. 1 with DHT4A nanoclay particles interspersed in a high density polyethylene (HDPE) polymer matrix. The challenge with such complex material images is that it is often difficult to distinguish between the polymer and clay particles as one cannot blindly search for any circular object. In this specific case the polymer has also formed bubbles at the edges of the polymer chains. An example of this phenomenon is highlighted in the green rectangle. Nanoclay particles that could be identified visually are circled with red circles. Apart from trying to distinguish between the two materials an added challenge is the agglomeration (grouping) of nanoparticles which results in a cluster (circled in blue on Fig. 1). In addition to these challenges the obtained SEM image itself may be compromised or changed during the imaging process due to sample preparation, choice of magnification level, differing environmental and lighting conditions [2]. These factors add a level of complexity to the analysis of SEM images, be it manual or automated. There is an obvious requirement to automate this tedious process.

The long term aim of this project is to propose an automated approach for the analysis of SEM images of composite materials. With the failure of considered techniques, a different approach will be developed.

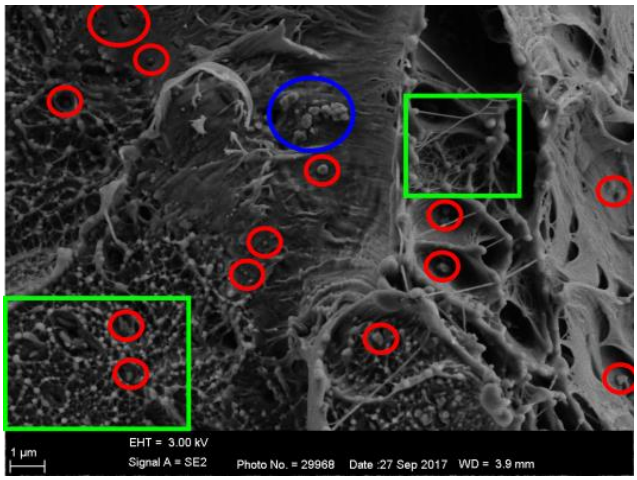


Fig. 1. SEM image of a HDPE-DHT4A nanocomposite material. Green rectangle illustrates polymer bubbles formed at the end of polymer chains and identified nanoclay particles are encircled with red circles with an agglomeration of clay particles shown in blue.

II. PRELIMINARY STUDY

Recent studies have started to address the automation of image analysis [2,5]. One of the challenges of automated digital image analysis is shape detection. The best machine

learning techniques for object detection are either training a classifier with a Convolutional Neural Network (ConvNet) [6-9] or the You Only Look Once (YOLO) method [10]. These techniques were implemented in an attempt to identify the presence of nanoparticles in the composite material SEM database with little to no success. It can be contributed to the small data set and the complexity of the images. These images are difficult to analyse manually and as such does not easily lend itself to an automated approach. When training the classifier on these images it was indifferent, as expected, in deciding whether clay was present in an image or not. In theory this method would be a good approach should a more comprehensive database be available to train a classifier on. The failure of the ConvNet and YOLO to identify the presence of particles in these SEM images steered the study back to exploring more traditional methods of identification which should be sufficient to identify particles in an image.

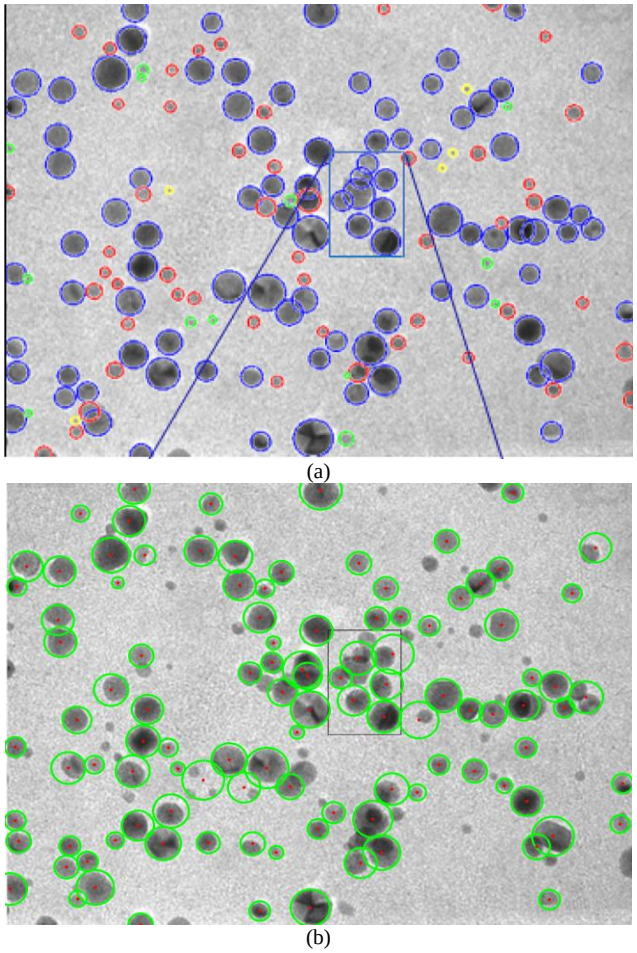
A circle detection technique that is popularly employed in image processing is the Hough transform (HT) [11] which is nearly 50 years old [12]. It is a robust method that also requires a lot of computational effort. The HT relies on various image pre-processing methods, including filtering and edge detection before identifying circular objects. However, as will be shown in Section III, the HT fails to successfully identify the nanoclay particles in the available composite material SEM image. For the purpose of demonstrating the proposed method a synthetic SEM image database [13] was used. These images only consisted of white powder particles on a black background. These images also mimic some of the complex particle behaviour seen in Fig. 1 where there are single particles of different sizes as well as agglomerated particles of various degrees of complexity. The HT also failed to successfully identify the majority of particles in the synthetic image and especially struggled to identify individual particles in the clusters.

As both the ConvNet and HT failed to identify particles in the composite and synthetic SEM images, the next step was to consider a different image processing methodology that will highlight (particle agglomeration and varying sizes) or suppress (background noise and polymer structure) complex features in the images to improve particle identification. It is proposed that by highlighting or suppressing key features the resulting images will assist the ConvNet in classifying the desired nanoclay particles. A combination of smoothing, filtering and contour approaches is considered for the proposed image processing methodology. For the initial development of the pre-processing method a synthetic SEM database [16] will be used in this study as the composite SEM images are too complex.

III. CIRCULAR HOUGH TRANSFORMS

Circular Hough Transforms (CHT) have been applied in studies to identify nanoparticles in TEM images [2,5]. To use CHT to identify nanoparticles, the methodology proposed by [2], was replicated and applied to a gold nanoparticle image from their paper. Once the proposed implementation was validated the methodology was applied to the more complex SEM image shown in Fig. 1 and finally on a synthetic particle SEM image.

[2] proposed an automated algorithm which comprised of a number of image pre-processing steps before applying a modified CHT to identify the nanoparticles. To replicate their approach it was decided to only make use of built-in image processing modules available in Python's OpenCV image processing library. The resulting detected circles using their approach and the replicated attempt are shown in Fig. 2(a) and 2(b) respectively, using the gold nanoparticles TEM image from [2].



(a)

(b)

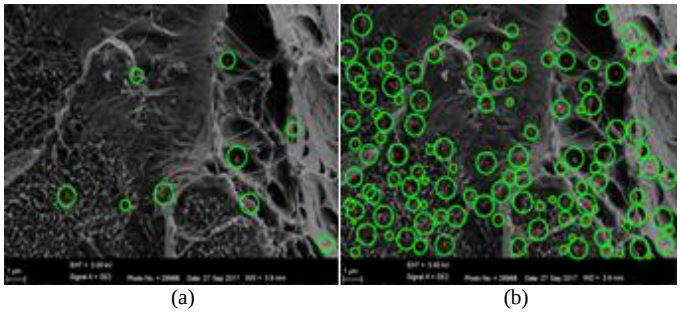
Fig. 2. Replication of the proposed methodology by [2] to identify nanoparticles using CHT. (a) Results from [2] and (b) replication attempt in this study on the gold nanoparticle TEM used in their study to develop the approach.

It is evident from Fig. 2 that the replicated attempt and proposed methodology is not as accurate as illustrated in [2], although it does manage to identify a large portion of the particles. The inaccuracy of the replication is likely due to differences in the mathematical formulations of the built-in Python modules, whereas [2] likely developed an in-house algorithm; and manually tuning the parameters for the various functions to replicate the visual results from their study as the paper did not provide any parameter or threshold values.

Nevertheless, the attempted replication is deemed suitable for the purposes of illustrating the use of CHT on a complex SEM image.

A. Application on composite SEM image

The results of applying this methodology to the more complicated SEM image from Fig. 1 are shown in Fig. 3. In Fig. 3(a) the same input parameters to the CHT in Fig. 2 were used, as was used for the gold nanoparticle TEM image when replicating the approach, whereas in Fig. 3(b) these parameters were manually adjusted in an attempt to identify more particles.



(a) (b)

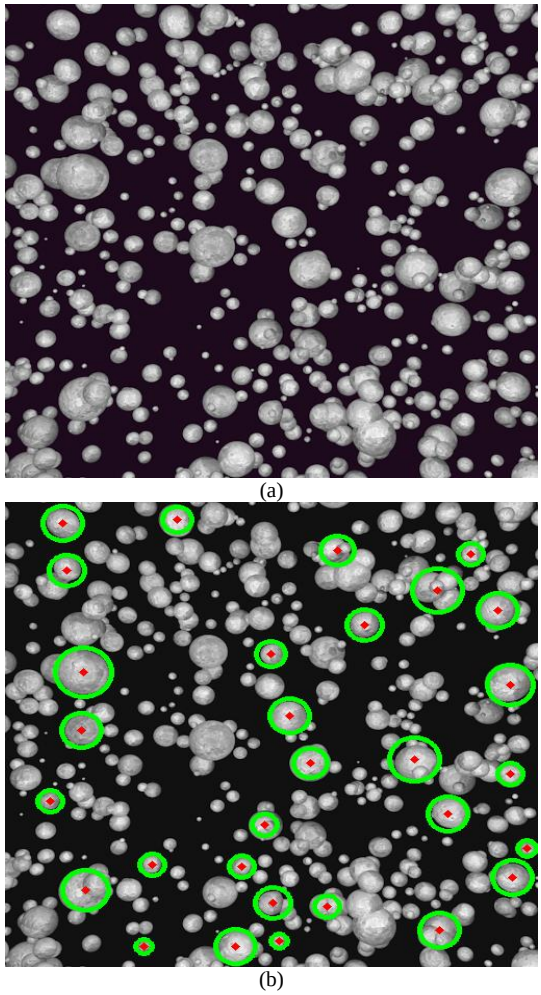
Fig. 3. Applying the CHT methodology to the SEM image of a HDPE-DHT4A nanocomposite material. (a) using the input parameters for the gold nanoparticle image and (b) manually tuning input parameters to obtain a better result.

It is clear from both results that the CHT is dependent on its input parameters and will be image specific, not a desirable trait when trying to automate a methodology, and is unable to identify the nanoclay particles of interest. These results show that CHT is not the best methodology for identifying nanoparticles in such complex images and that a different approach to processing the image is required. In addition, these images are too complex when trying to establish a methodology to identify particles.

B. Application on synthetic SEM image

A synthetic database of 2048 powder material SEM images was constructed by [13]. The images were generated using open source rendering software and used eight particle size distributions that are similar to one another. This allows the images to be used as a benchmark when comparing, testing or developing image processing techniques. For this paper one image from this dataset will be used during the development of the proposed image processing technique. The image was randomly chosen and used a Weibull probability distribution (μ=0.1, σ=0.5) when generating the particles. The synthetic SEM image is shown in Fig. 4(a) and the detected particles when applying the CHT methodology from [2], with input parameter tuning, is shown in Fig. 4(b). This result reiterates the need for a different image processing technique to identify the particles due to the complex nature of the particle interactions, e.g. agglomerated particles and particles shadowing neighbouring particles. This synthetic SEM image will be used for this preliminary study.

Using Hough Transforms to identify circular shapes can be very complicated, as illustrated in this section. It is concluded that this method would not be able to clearly identify particles in a synthetic or complicated SEM image. Whether a different pre-processing method will improve object recognition methods of particles in an image is yet to be determined.



(a)

(b)

Fig. 4. (a) Synthetic SEM image of a powder material using a Weibull probability distribution for the particles and (b) CHT methodology applied to the synthetic SEM image with 30 particles identified.

Due to the large quantity of synthetic particles in the images it is too complicated and labour intensive to count the number of particles in each image, which is used to quantify the method performance. The solution is to use a randomly selected region in a randomly selected image to validate the proposed method.

IV. IMAGE PROCESSING OF SYNTHETIC SEM IMAGES

A method of pre-processing on an image in order to identify object shapes that has not been considered in previous literature is to create a contour of the original image in order to identify circular objects. This could also deal with occluded and partially visible particles. In an image with complicated structures, contours can be used to highlight circular shapes which should make it easier for a ConvNet algorithm to identify the particles. This method can enable a user to select

and display certain sized contours, which represents the radiuses of typical particles in the image. The method for highlighting particles using the contours is developed and tested using the synthetic SEM image database.

A. *Contours and Bounding Boxes*

Using Fig. 4(a) for the development of this method, the contours were extracted using a Python contour plot function and the results are shown in Fig. 5. The contour plot function uses the coordinates of the height values (particle boundary intensity) and the height values as input to represent the 3-dimensional surface in 2-dimensions. The contours, when extracted, are sorted into layers/levels.

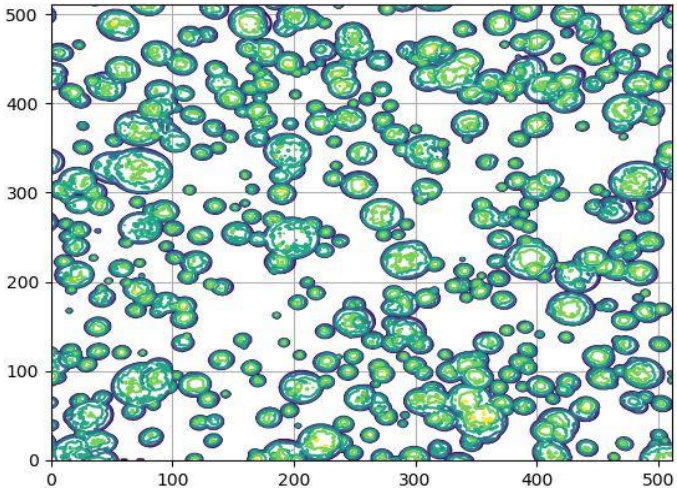


Fig. 5. Contour plot of the synthetic image.

The efficiency of simplifying an image using contours is determined by utilising the contours to count the amount of particles in the image. Each particle is identified by using bounding boxes on the contours. The highest layer of contours, which corresponds to the highest peaks in the image, are selected and the center point for each is calculated by taking the average of the horizontal and vertical coordinates of the respective contours. The second highest contour layer is selected and compared with the identified center points (from the highest layer) to determine if any of these points fall within the contours. In the case where contours are not associated with any points, the center point is calculated and added to the existing array of center points. This is repeated for each of the layers. This point array is considered the highest intensity in each particle; including individual particles clustered together, see Fig. 6, where the highest intensity is denoted with a red 'x'.

The original image is digitised and the true center position of each particle is manually determined. These coordinates are indicated with a blue 'x'. It is clear from Fig. 5 and 6 that there are local maxima and minima in the intensity, resulting in more than one maximum on the particles. These local maxima and minima add unwanted complexity to the image and to minimise this, image filtering and smoothing techniques was applied.

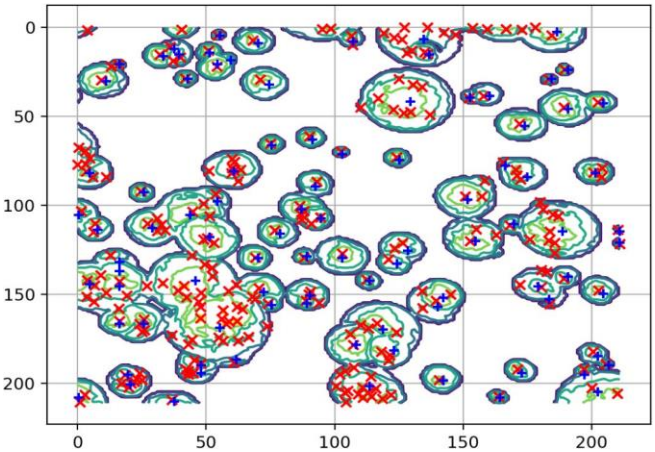


Fig. 6. Zoomed contour plot of the synthetic image with a red 'x' identifying the highest points of particles and blue 'x' identifying the true position of the particles.

B. *Filtering and smoothing*

The initial filtering of the image is done by calculating the average intensity over the whole image and setting all pixels, with intensity values below the calculated average threshold, to zero. Before any smoothing is done the contours are sorted so that all the contours falling in the same coordinate range are grouped to represent the contents of each particle and cluster. The contour groups that have only 1 contour for every layer/level are identified and saved. This is done so that the coordinates of very small particles, which could vanish during more intensive image processing such as filtering and smoothing, are identified.

Two methods are considered for image smoothing, namely a convolution and LULU [14,15] smoothing, which will be discussed in the following paragraphs. Finally after smoothing a multidimensional median filter is used, using a Python function called *scipy.ndimage.filters.median_filter* with a footprint or filter of 5. This function denoises the smoothed image a final time. Finally the smoothed contours of each particle are extracted from the image. The bounding boxes method (described in Section IV.A.) is used, combined with the list of single particles identified before smoothing, to identify the number and position of the particles using the contours and function as a accuracy metric.

The convolutional smoothing was done over the horizontal and vertical axis of the image. The horizontal axis smoothing was performed by iterating over the vertical axis of the image, effectively taking 2 dimensional slices over the horizontal axis. Each peak intensity or cluster of peaks, in each slice, are smoothed separately and then stitched into the image. This iterative procedure is repeated for the vertical axis by using the horizontal axis smoothed image as input. For the smoothing the Python function *numpy.convolve* was used which returns a linear convolution of two one-dimensional arrays. By setting one array as the data slice to smooth and the second as an array with a length of the number of pixels to smooth over; and an intensity of 1/length, the result is a smoothed output. The length is chosen as 5. If the length of the particle or

agglomerated structures is less than 5 the length is set as two less than the length. Setting the mode to 'same' ensures that the boundary effect is still visible and important information is not lost. The contour plot of the image with convolutional smoothing and a median filter is displayed in Fig. 7, with the particles identified with red crosses using bounding boxes and the blue 'x' the particle's true center coordinates.

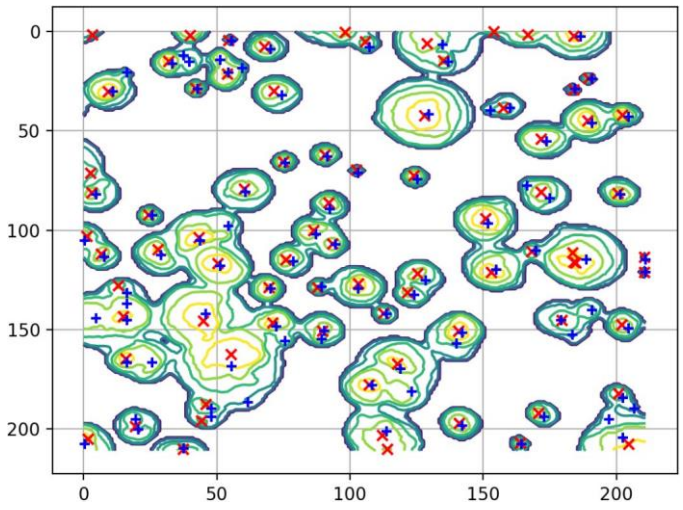


Fig. 7. Contour of image with convolutional smoothing over the horizontal and vertical axis and median filter, with the red 'x' indicating the particle position identified using bounding boxes and blue 'x' show the true center position of each particle.

The second method considered for image smoothing is LULU smoothing [14]. LULU smoothers operate on the repeat application of minimum and maximum operators on a moving window on the data. The two base operators are the L ("lower") and U ("upper") operators.

Consider a 1 dimensional series **X** where an operator with a window size n will be applied to element $x_j \in \mathbf{X}$:

1. Create $n+1$ sequences, each containing x_j.

$$\begin{aligned} S_0 &= (x_{j-n}, \ldots, \boldsymbol{x_j}), \\ S_1 &= (x_{j-n+1}, \ldots, \boldsymbol{x_j}, x_{j+1}), \\ &\vdots \\ S_{n-1} &= (x_{j-1}, \boldsymbol{x_j}, \ldots, x_{j+n-1}), \\ S_n &= (\boldsymbol{x_j}, x_{j+1}, \ldots, x_{j+n}). \end{aligned} \quad (1)$$

2. $$L(x_j) = \max(\min(S_0), \cdots, \min(S_{n-1}), \min(S_n)) \quad (2)$$
$$U(x_j) = \min(\max(S_0), \cdots, \max(S_{n-1}), \max(S_n)) \quad (3)$$

For example: The L and U operator for a 1 dimensional series with $n = 1$ will be:

$$S_0 = (x_{j-1}, \boldsymbol{x_j}),\ S_1 = (\boldsymbol{x_j}, x_{j+1}), \quad (4)$$

$$L(x_j) = \max(\min(S_0), \min(S_1)),\ U(x_j) = \min(\max(S_0), \max(S_1)).$$

These operators can be applied in a variety of different configurations and sequences, depending on the data that requires smoothing. The name of the smoother is an example of how the operators can be applied. LULU implies L was first applied, followed by U, L and U.

The advantage of LULU smoothing is the ability to maintain the features of the dataset while filtering out the noise. Consider Fig. 8 where the L, U and LU operators as well as a Moving Average trendline is applied to the data (denoted by blue **x**). Applying LU to the dataset removes all the outliers and/or noise, while the moving average adds features that do not exist in the original dataset. Visible in Fig. 8 is that L(X) produce a series that includes any outliers that tend to fall below the dataset's trend of the data immediately surrounding the outlier. The opposite is true for U(X). LU(X) combines the best of both L(X) and L(Y) and results in a series that excludes all the outliers introduced in X. It should be noted that the LU operator was specifically chosen to remove the noise from this dataset and might not work on another set.

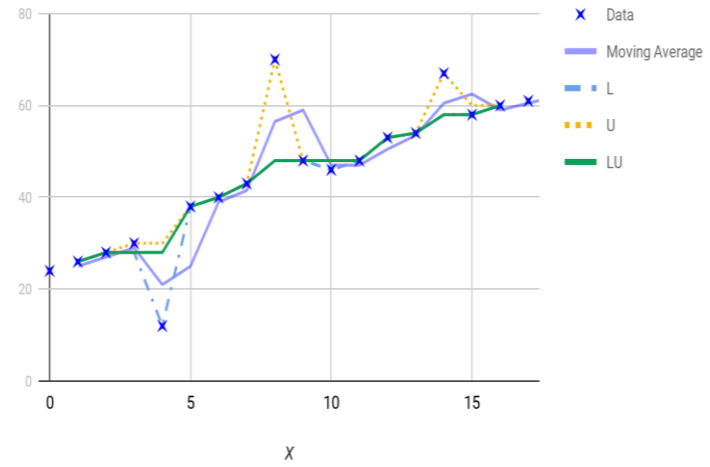
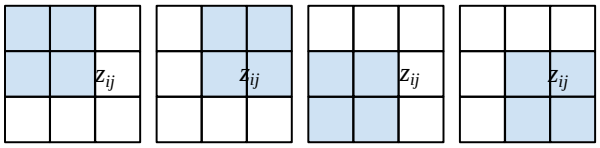


Fig. 8. L, U and LU operators applied to a generated data set, compared with a moving average trendline with a width of period of 2.

To apply LULU operators to a 2 dimensional image will only require adjustment of the window design [15].

For example: L and U operators for a 2 dimensional series **Z** with n=1.



$$\begin{aligned} S_0 &= (z_{i-1,j-1}, z_{i-1,j}, z_{i,j-1}, \boldsymbol{z_{i,j}}), \\ S_1 &= (z_{i,j-1}, \boldsymbol{z_{i,j}}, z_{i+1,j-1}, z_{i+1,j}), \\ S_2 &= (z_{i-1,j}, z_{i-1,j+1}, \boldsymbol{z_{i,j}}, z_{i,j+1}), \\ S_3 &= (\boldsymbol{z_{i,j}}, z_{i,j+1}, z_{i+1,j}, z_{i+1,j+1}). \end{aligned} \quad (5)$$

$$L(z_{ij}) = \max(\min(S_0), \min(S_1), \min(S_2), \min(S_3)) \quad (6)$$

$$U(z_{ij}) = \min(\max(S_0), \max(S_1), \max(S_2), \max(S_3)) \quad (7)$$

The effectivity of a specific operator combination is determined by counting the number of identified particles ('x') and calculating the percentage with respect to the number of actual particles in the image. This is an approximate metric to narrow a large search space and does not address mislabeled or missed particles. Examples of combinations with the

accuracy metric are listed in Table I. For this metric, lower percentages indicate better correlation between the true number of particles and the estimated number. A large number of operator combinations were tested of which only the combinations considered are displayed in Table I.

TABLE I. DIFFERENT SMOOTHING LU OPERATORS AND COMBINATIONS

Operator combinations	**Total number of true particles counted 489**	
	Number of particles identified	***Accuracy metric***
LU2 in column (no median filter)	570	16.6%
LU2 with median filter	465	4.9%
ULUL2 in row and column (no median filter)	460	5.9%
L with median filter	429	12.3%
UL2 in row and column (no median filter)	460	5.9%
U2 with median filter	449	8.2%

The operator combinations considered for smoothing are U2 with the median filter, UL2 and ULUL2 in the row and column with no median filter. These operator combinations are selected by consulting the metrics displayed in Table I as well as visual inspection. LU2 is not considered a suitable method even though it has the best accuracy, with an accuracy of 4.9%. The method is eliminated due to a boundary effect, distorting one edge of the image, which skews the result. The contour plot of an image with U2 smoothing and a median filter is displayed in Fig. 9; with the particles identified by red a 'x' using the bounding boxes method.

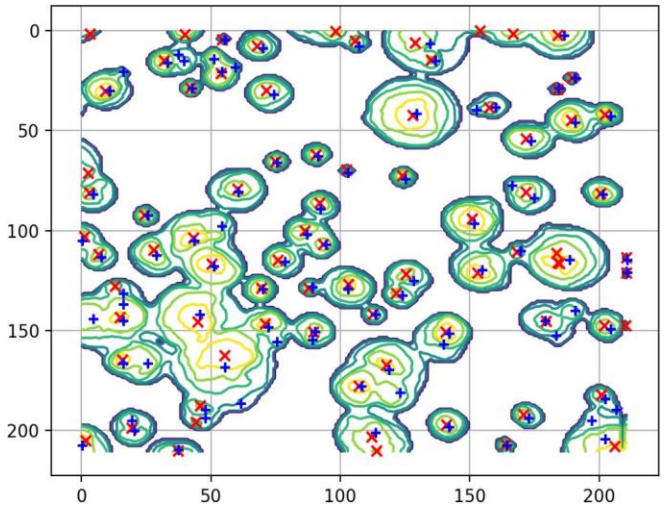


Fig. 9. Contour of image with U2 smoothing, 'x' indicates the position of each identified particle.

To quantify the most efficient smoothing method, which minimise the local minima and maxima on a particle while not suppressing important features, the true particle positions are compared, in Table II, to the position and number of particles in the smoothed images which were identified using the contour and bounding boxes method. Due to the size and complexity of the images only pixel 0 - 220 is looked at in the horizontal and vertical axis. In Table II the column named 'Number of 'x'' is the number of coordinates identified using smoothing, contours and bounding boxes. The second column 'Number correct 'x'' lists the number of particles, correctly identified, while 'Number wrong 'x'' is the number of 'x'es that misses a particle or when more than one 'x' appear on a particle. 'Number particles missed' is the number of particles not identified using the proposed method, while the final column 'Accuracy metric' is calculated by:

('Number correct 'x'' - ('Number wrong 'x' particles'

+ 'Number particles missed'))/Total true particles, (8)

(75 - (7+16))/92 = 0.565. (9)

TABLE II. PARTICLE IDENTIFICATION WITH DIFFERENT SMOOTHING METHODS

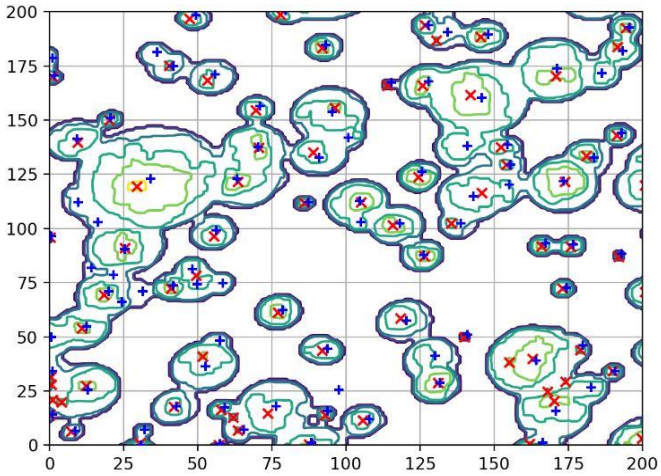
Smoothing Method	**Total number of true particles counted = 92**				
	Number of 'x'	***Number correct 'x'***	***Number wrong 'x'***	***Number particles missed***	***Accuracy metric***
No smoothing	243	92	151	0	-0.641
Convolution	75	71	4	21	0.5
U2	76	71	5	21	0.489
UL2	83	75	7	16	0.565
ULUL2	83	75	7	16	0.565

From Table II, the UL2 and ULUL2 smoothing performs the best, with the most correctly identified particles and the least number of particles missed. While the number of wrong 'x'es are higher compared to the other methods, the performance for the number of correctly identified particles and missed particles outweigh this, as can be seen from the accuracy metric in the last column. It was found that these methods remove the most local minima and maxima in the intensity while preserving the important features for particle identification. The UL2 and ULUL2 results are identical since the ULUL2 smoothing is the result of applying UL2 twice. Because the second round of smoothing has very little effect, the results are almost identical. The UL2 is selected as the optimal smoothing method for the proposed technique.

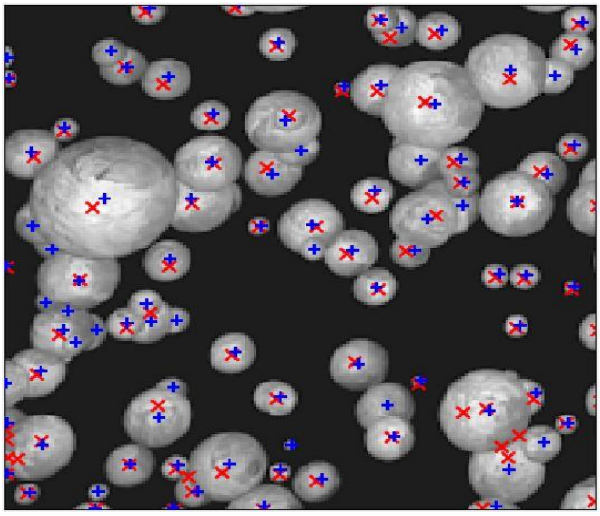
C. Results

The method developed in Section IV, using UL2 smoothing, was then applied on a previously unseen image shown in Fig. 10(a) and the actual figure with 'x' indicating particle centers is shown in Fig. 10(b). The total number of particles manually counted is 91. The total number of 'x'es obtained from the smoothing and contours are 77, while 70 are correctly labeled. 7 'x'es are incorrect and 21 particles were missed. The accuracy is calculated as 0.462. This score is not as good as the score for the image discussed in Section IV.B, which is not surprising due to the complexity in the randomly selected

image, shown in Fig 10(b), with many particles only partially visible.



(a)



(b)

Fig. 10. (a) Smoothed contour plot with each particle identified with a red 'x' and (b) the original SEM image with each particle identified by a red 'x'.

V. CONCLUSION

The UL2 and ULUL2 smoothing methods gave the best result, but all methods listed in Table II performed relatively well on the first image. The proposed method performs reasonably well on a previously unseen complex image compared to the image on which the method was developed. This implies that this technique is suitable for automation and can therefore be used on a large dataset. The extent of filtering and smoothing as well as types of smoothing have to be customised for different types of images, but the technique is very simple to implement and mostly uses built-in Python functions. In addition, the method used to quantify the smoothing efficiency, employing contours in conjunction with the bounding boxes, succeeded in identifying more particles than the Python built-in Hough transform. The proposed pre-processing technique was able to largely highlight the features of interest and make it possible to identify single particles as well as individual particles in agglomerated structures. It is concluded that this technique, with some tailoring, may be suitable for highlighting the needed features to assist a ConvNet in the identification of particles.

In future studies the proposed technique will be applied to the larger synthetic SEM database to add another feature or dimension for use in an automated ConvNet approach for particle identification. Once the automated algorithm has been developed and tested on the synthetic database, this approach will be extended to the composite SEM image of interest.

REFERENCES

[1] J. Jordan, K.I. Jacob, R. Tannenbaum, M.A. Sharaf, I. Jasiuk, 2005, Experimental trends in polymer nanocomposites – a review, Materials Science and Engineering A, vol. 393, pp. 1 – 11.

[2] Mirzaei, M. and Rafsanjani, H.K., 2017. An automatic algorithm for determination of the nanoparticles from TEM images using circular hough transform. Micron, 96, pp.86-95.

[3] B.L. DeCost and E.A. Holm, 2017, Characterizing powder materials using keypoint-based computer vision methods, Computational Materials Science, vol. 126, pp. 438 – 445.

[4] A. Nanakoudis, 2018, SEM and TEM: what's the difference?, ThermoFisher Scientific Blog (http://blog.phenom-world.com/sem-tem-difference), viewed 16 October 2018.

[5] Meng, Y., Zhang, Z., Yin, H. and Ma, T., 2018. Automatic detection of particle size distribution by image analysis based on local adaptive canny edge detection and modified circular Hough transform. Micron, 106, pp.34-41.

[6] Uijlings, J.R., Van De Sande, K.E., Gevers, T. and Smeulders, A.W., 2013. Selective search for object recognition. International journal of computer vision, 104(2), pp.154-171.

[7] Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[8] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[9] Dai, J., Li, Y., He, K. and Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems (pp. 379-387).

[10] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[11] Mukhopadhyay, P. and Chaudhuri, B.B., 2015. A survey of Hough Transform. Pattern Recognition, 48(3), pp.993-1010.

[12] Hough, P.V., 1962. Method and means for recognizing complex patterns. U.S. Patent 3,069,654.

[13] DeCost, B.L. and Holm, E.A., 2016. A large dataset of synthetic SEM images of powder materials and their ground truth 3D structures. Data in Brief, 9, pp. 727-731. (https://data.mendeley.com/datasets/tj4syyj9mr/1)

[14] Rohwer, C.H., 1989. Idempotent one-sided approximation of median smoothers. Journal of Approximation Theory 58, 151–163. https://doi.org/10.1016/0021-9045(89)90017-8

[15] Rahmat, R., Malik, A.S., Kamel, N., 2013. Comparison of LULU and Median Filter for Image Denoising. International Journal of Computer and Electrical Engineering 568–571. https://doi.org/10.7763/IJCEE.2013.V5.774