On the Use of ImageJ Segmentation

(Article begins on next page)

28 April 2024

On the Use of ImageJ Segmentation

A. C. Sparavigna[1]

[1] Dipartimento di Scienza Applicata e Tecnologia, Politecnico di Torino, Torino, Italy

Many methods for the image segmentation are based on thresholding, that is on the conversion of images into black and white ones. Actually, after the thresholding, it is easy to identify the black or white domains and calculate the features of them. Of course, it is necessary to write instructions for the processing of a digital image into a programming language. This task could be beyond the skills of the user or the scope of user's investigation. In this case, a method based on software ImageJ can be the simplest approach for the considered research. However, this method needs to be properly applied.

Introduction In the image processing a segmentation is a method of partitioning an image into multiple segments, which consist of sets of pixels. These segments are also known as image objects or super-pixels. Many can be the goals of segmentation; basically they consist in extracting features by the image representation into the set of segments. Among the image segmentation methods, we have also those which are devoted to the determination of the contours of objects; in this case, the segmentation is known as an edge detection (on image segmentation and edge detections, see Refs.1-14).

Many methods of segmentation are based on the thresholding of the image. That is, the image - made by grey pixels for instance - is converted into a black and white image, by choosing a specific value of grey (the threshold). The pixels having a grey tone with a value larger than that of the threshold are converted into white tones, whereas the pixels having a grey tone lower or equal to the threshold are converted into black pixels. Once the image is converted in this manner, the black or the white domains, that is the super-pixels, can be identified, that is labelled, by means of an integer or a colour. Of them we can evaluate centre, area, perimeter and other geometrical features, as explained in [15,16]). In Ref.15 it is explained the process of labelling the super-pixels.

An approach based on thresholding is very useful for investigating the images of vesicular textures,

that is the textures that we can observe in materials having inside a distribution of pores [15]. In particular, in 2017 the method was applied for the measurement of the pore size distribution of freeze-dried solutions, as observed by means of images obtained by the scanning electron microscopy [17,18]. The presence of pores is due to ice crystals. The results of the segmentation technique show a good agreement with a model prediction of the original ice crystal size distribution, based on balance equations.

Other applications are given in [19-29]. In Refs. 16-29, the segmentation was obtained by the proper logic instructions written in Fortran 77 programming language. The result of the segmentation was rendered in images where each super-pixel has a different colour (see Figure 1).
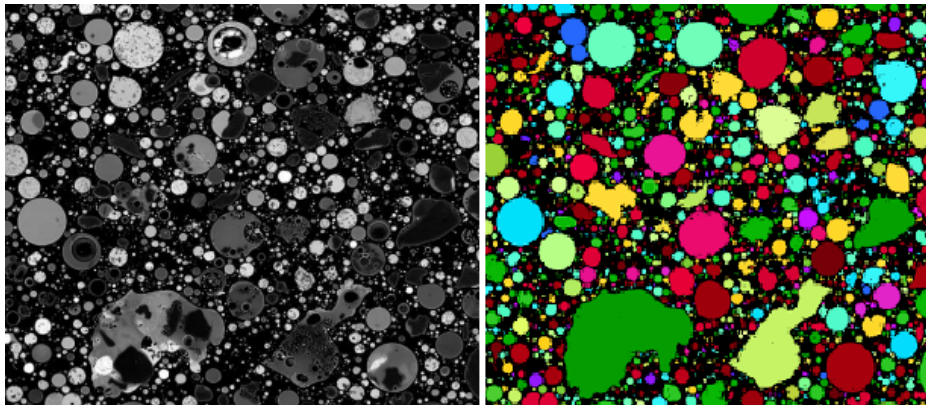

Figure 1: On the left a fly ash micrograph (courtesy Wikipedia and user Wabeggs), made by means of a scanning electron microscope (SEM) and back-scatter detector. On the right, the related image segmentation (see Ref.22).
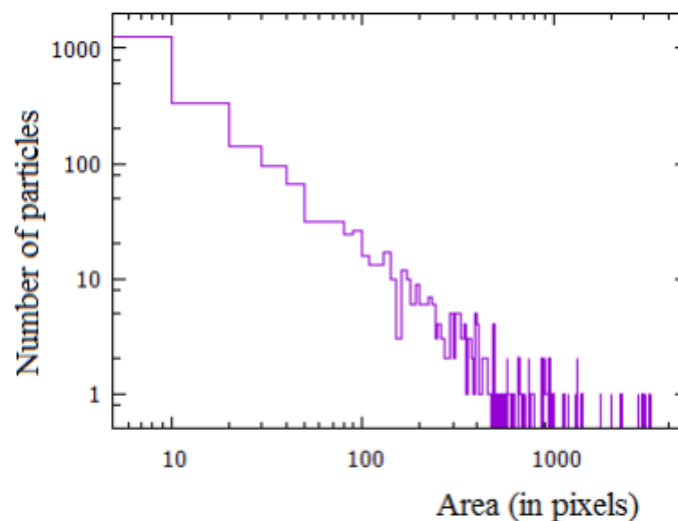

Figure 2: Distribution of the cross-sections of particles (we have not considered the two largest irregular particles (green and pale green) of the Figure 1. The area are spaced in intervals of 10 pixels.

After the segmentation, we can analyse the distribution of particles. For instance, we can obtain the distribution of the cross-sections of the particles of the Figure 1, and obtain the result given in the Figure 2.

The "everything has been done" trap Of course, the best condition for any image segmentation is that of having a program, in which the instructions can be changes and adapted to the specific problem under investigation (that was the case for the studies proposed in Refs.15-29). However, to write the instructions for the processing into a programming language is a task that could be beyond the skills of the user or the main scope of user's investigation. In this case, a method based on software, such as ImageJ, can be the simplest approach for the considered research, extracting information about the features of super-pixels from the set of measurements that software is proposing. Some references on ImageJ are [30-47].

The number of publications on ImageJ and related tools is huge and it continues to grow. So it seems that any work on image segmentation can fall into the "everything has been done" trap. This is not true. Let us stress that, first of all, it is important to understand the methods of segmentation, even the starting point of it, that is the thresholding, such as that used in ImageJ, and the labelling of segments. On the basis of [15], we can imagine different approaches to thresholding and labelling; these steps are depending on the objects that we are considering and studying. As it is easy to understand, a large difference in the results can be produced by a different thresholding, then a tip to get past the "everything has been done" trap is to perform a different thresholding and analyse the different results that we can obtain, because the use of ImageJ is a part of any study that we can make on images and not the whole.

ImageJ Let us consider the use of software ImageJ, with the image on the left in the Figure 3. It is a courtesy of Dr. Mae Melvin, USCDCP, showing a photomicrograph of a blood smear with erythrocytes and malaria plasmodium parasites. After the image is uploaded, the first processing that we have to make is its thresholding. The suggested approach is that of change the colour image into an image having pixels in grey tones, and upload the grey-tome image, such as that in the middle of the Figure 3. The grey-tone image has been obtained by means og GIMP, the GNU Image Manipulation Program. Then, let us make the thresholding of this image, by means of the ImageJ instruction sequence Image > Adjust > Threshold. The result is given on the right of the Figure 3, for the default value of threshold. Let us note that we can make the thresholding using GIMP too. Moreover, a preprocessing of the image is often required, before any segmentation process we can

imagine. The micrograph in the Figure 3 on the left is a special case, because it is not requiring any specific preprocessing for being segmented.
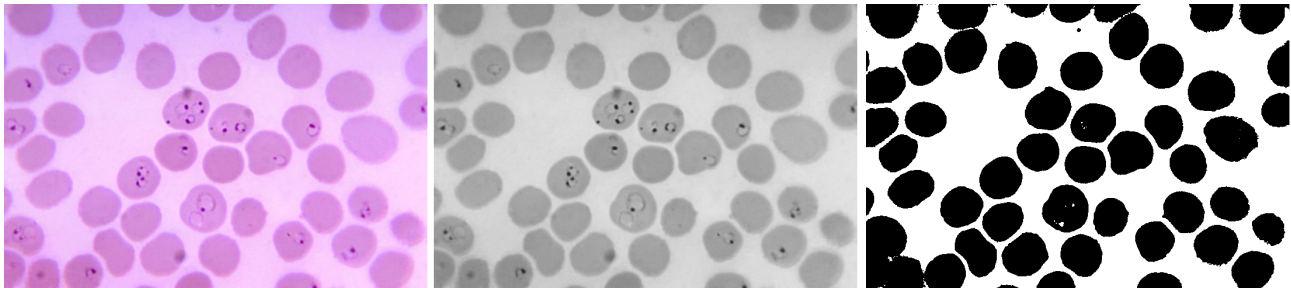

Figure 3.

After the black and white has been obtained, we have to make it binary for ImageJ. We use the following sequence of instructions: Process > Binary > Make Binary. That is, we make it truly binary for ImageJ. We have the image that software can analyse. The segmentation is obtained by means of the following sequence of instructions: Analyze > Analyze Particles. The result is given in the Figure 4. The domains on edges can be excluded as in the given image. Each domain is shown by means of an outline (but other options are possible). It is labelled by means of an integer, as we can see in the Figure 4. In a panel, which can be saved in a file, corresponding to each label, we can find the area of the super-pixel and the coordinates of its centre.
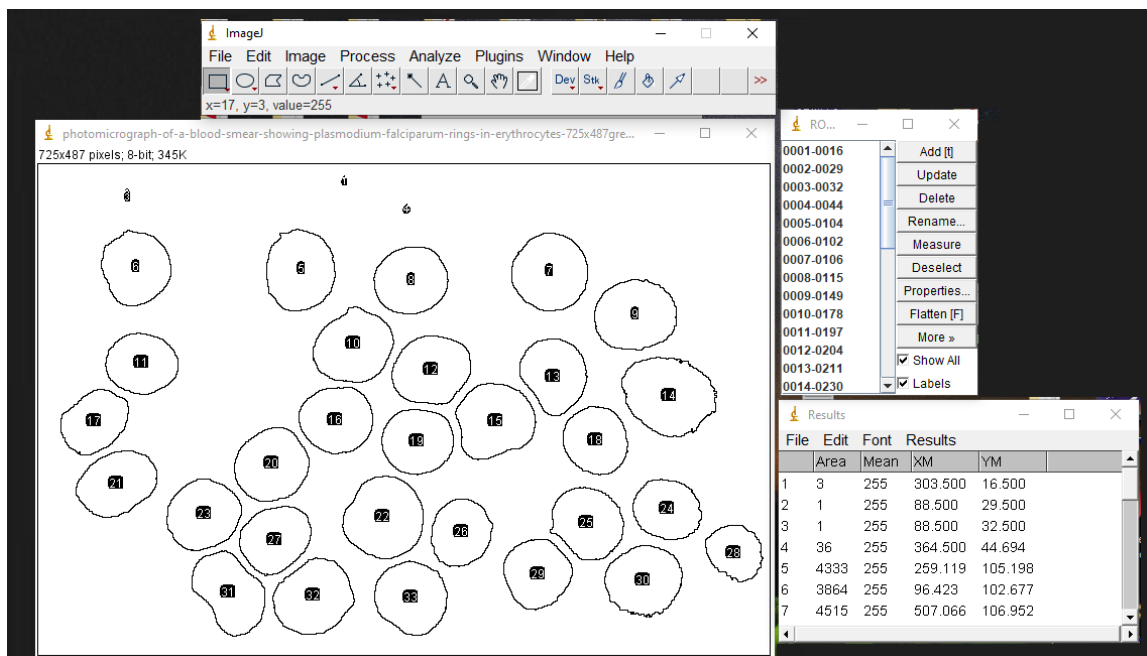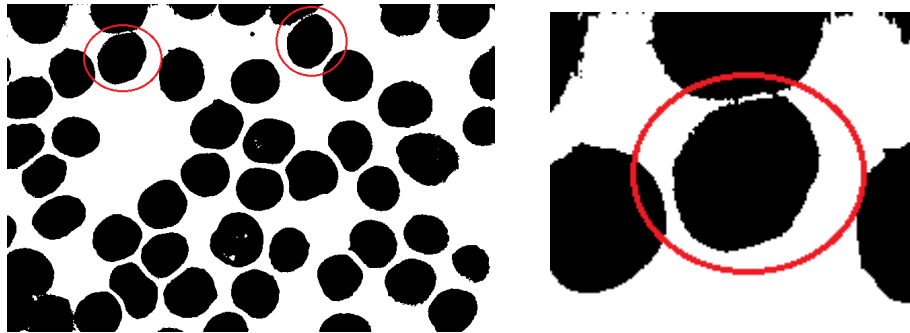

 Figure 4.

Figure 5.

As we can see in the Figure 4, two large particles are not counted. That is the number of cells is 30 instead of 32. The count is affected by an error of 6%. The particle not counted are shown encircled in red in the Figure 5. The problem is that two black pixels are joining the cell to that which is above. We can try to adjust the value of the threshold in order to reduce the area of the objects, and have in this manner, the cells separated. However, a different thresholding method or a further control on the labelling of the particles will help to avoid such a problem.

Colour segmentation If we try to apply the Particle Analyzer on RGB images, a panel tells us the these images must be thresholded using Image > Adjust > Color Threshold. So, in the case of the Figure 2 on the left, let us try to make the segmentation of the colour image. After adjusting by means of the Color Threshold, with the default method, we can obtain the segmentation as in the Figure 7.
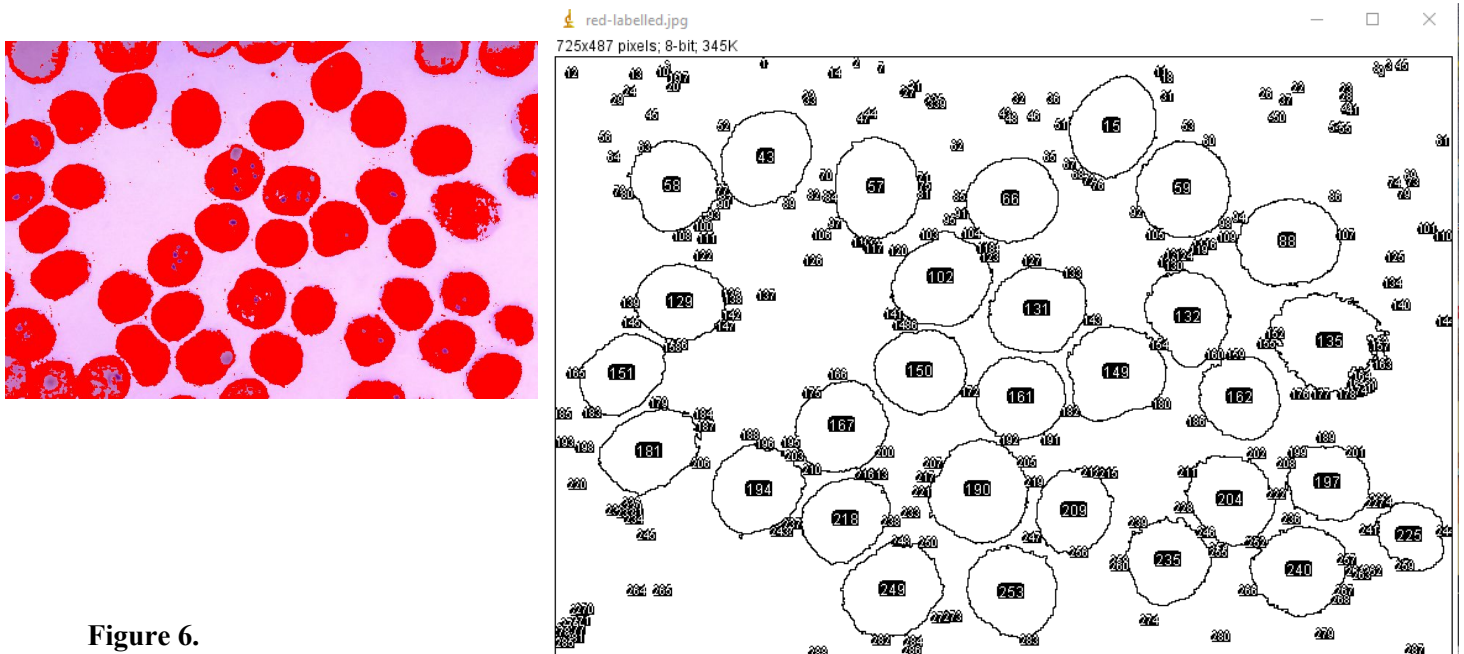

Figure 6.

The two large domains, that in the previous approach (see Figure 5) have not been counted, are visible in the colour segmentation of the Figure 6. However, in the Figure 6, we can see that we have also a lot of small domains. These domains can be relevant or not, but this is depending on the specific features that we have to consider.

In the segmentations that we have shown, the cells containing the plasmodium parasites are considered as equal to those which are not infected. Therefore, a different processing is required to determine the presence of the parasites. For what concerns the automated detection of malaria from blood smear, and other related problems, let us suggest Ref.48.

Different thresholds The ImageJ Threshold Color possesses several thresholding methods. Among them we find the Default, MaxEntropy, Otsu, and so on. The results that we obtain are different, besides being dependent on the value of the threshold.

For instance, if we use the Otsu method of thresholding, an image that we can use for the segmentation is like in the Figure 7.
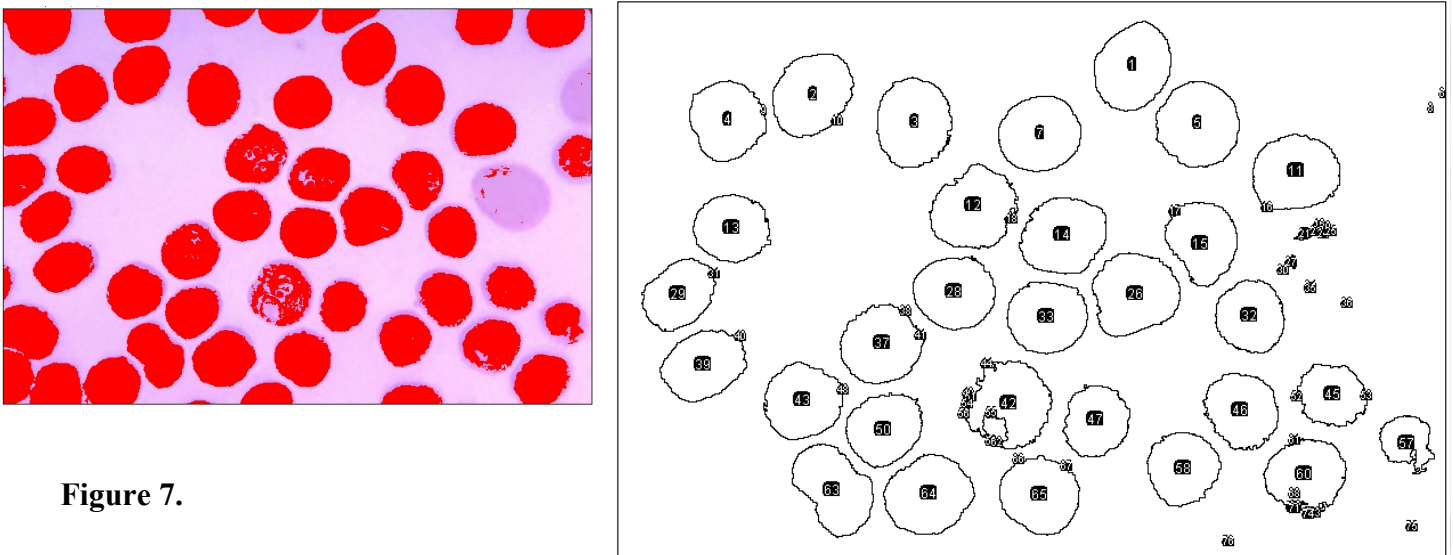

Figure 7.

Again, we can see that one of the particle is lost in the count. And the areas occupied by the particles are different.

About the thresholding, the web page https://imagej.net/Thresholding#FAQ (http://archive.ph/mM3DB) is proposing the following question: "How do I know whether my threshold is correct?", and the answer is this: "In short, you can't. It will always be, to some extent, in the eye of the user/observer/scientist and will also be impacted by empirically collected knowledge. The basic problem of deciding if a threshold (or in general an extraction method) is

"good" needs a "ground truth". But such a ground truth is not naturally existing and is always created in one or the other way by a human. So, describing which method you use—and/or showing a comparison with other methods—is probably the best you can do to enable a statement on the quality of the extraction."

Directions For what concerns the measurements that ImageJ software is providing, they are available using Analyze > Set Measurements. Let us consider a specific measurement (we can find all the details at the following link: https://imagej.nih.gov/ij/docs/menus/analyze.html)- And it is the following: we want to determine the directions of rod-like particles. In [27,28], the approach was based on the calculus of momenta of inertia, and the R-language has been used. Here, let us use ImageJ and the command Fit Ellipse, which is rendering a rod-like object as an ellipse. At the link given above, we find the following detail: "Fit Ellipse - Fit an ellipse to the selection. Uses the headings Major, Minor and Angle. Major and Minor are the primary and secondary axis of the best fitting ellipse. Angle (0-180 degrees) is the angle between the primary axis and a line parallel to the x-axis of the image. The coordinates of the center of the ellipse are displayed as X and Y if Centroid is checked. Note that ImageJ cannot calculate the major and minor axis lengths if Pixel Aspect Ratio in the Set Scale dialog is not 1.0." An example.
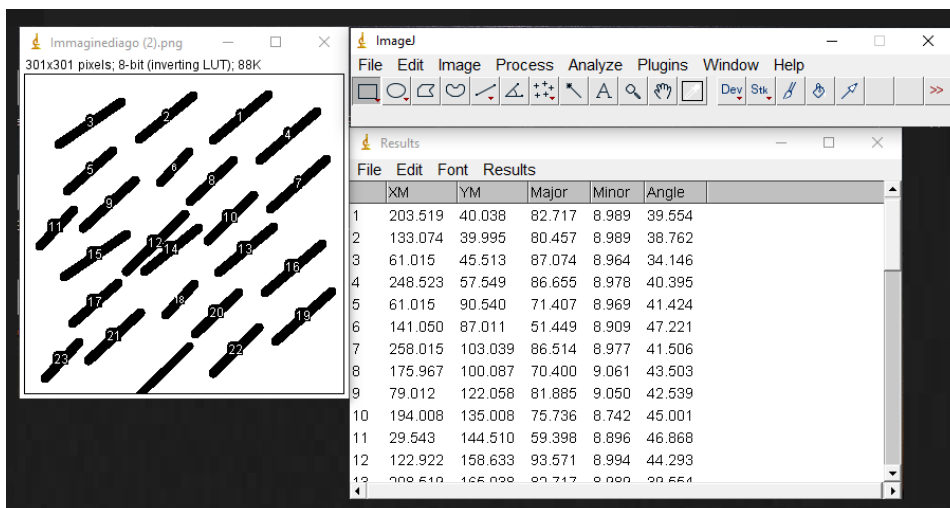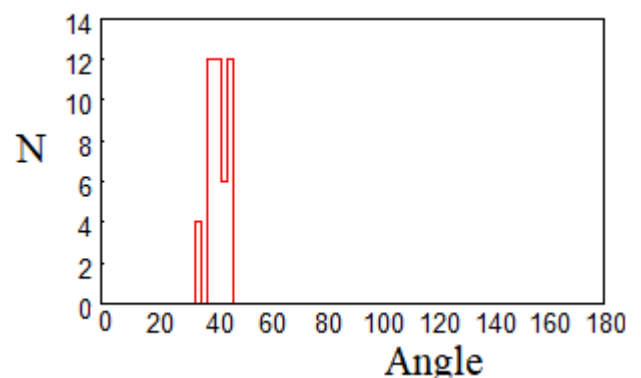

Figure 8.

Figure 9.

We can use the data provided by ImageJ to have the angular distribution of the rod-like object (Figure 9). The result of the distribution depends on the chosen angular interval.

Discussion The two images that we have used here are quite "good" images. Usually, it is not so. So we have to be very careful in applying software and the several methods it is offering.

Actually, ImageJ is a powerful software for the segmentation of image. However, and this is true for any software based on thresholding that we can use, the result depends on the threshold that we decided to have the binary image. Then, the final results, such as the number of particles and other related measurements, are linked to the value of the threshold. The example proposed in the Figures 4 and 6 is clearly showing this fact. Therefore we have to estimate carefully the uncertainty of the count of particles, in particular when this uncertainty is larger than that requested by the case study. Let us stress once more that, because of the dependence on thresholding, we have not to be surprised that measurements can change depending on the preprocessing of the image. The difference, in percentage, can be relevant or not and this depends on the problem under study.

References

[1] Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. Pattern recognition, 26(9), 1277-1294.

[2] Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. Computer vision, graphics, and image processing, 29(1), 100-132.

[3] Sharma, N., & Aggarwal, L. M. (2010). Automated medical image segmentation techniques. Journal of medical physics/Association of Medical Physicists of India, 35(1), 3.

[4] Zaitoun, N. M., & Aqel, M. J. (2015). Survey on image segmentation techniques. Procedia Computer Science, 65, 797-806.

[5] Khan, W. (2013). Image segmentation techniques: A survey. Journal of Image and Graphics, 1(4), 166-170.

[6] Varshney, S. S., Rajpal, N., & Purwar, R. (2009, December). Comparative study of image segmentation techniques and object matching using segmentation. In 2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS) (pp. 1-6). IEEE.

[7] Zuva, T., Olugbara, O. O., Ojo, S. O., & Ngwira, S. M. (2011). Image segmentation, available techniques, developments and open issues. Canadian Journal on Image Processing and Computer Vision, 2(3), 20-29.

[8] Marr, D., & Hildreth, E. (1980). Theory of edge detection. Proceedings of the Royal Society of London. Series B. Biological Sciences, 207(1167), 187-217.

[9] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and

machine intelligence, (6), 679-698.

[10] Torre, V., & Poggio, T. A. (1986). On edge detection. IEEE Transactions on pattern analysis and machine intelligence, (2), 147-163.

[11] Scherer, R. (2020). Feature Detection. In Computer Vision Methods for Fast Image Classification and Retrieval (pp. 7-32). Springer, Cham.

[12] Goodsitt, J. E., Walters, A. G., Abad, F. A. T., Taylor, K., Farivar, R., Pham, V., & Truong, A. (2019). U.S. Patent Application No. 10/311,577.

[13] Liu, X., Deng, Z., & Yang, Y. (2019). Recent progress in semantic image segmentation. Artificial Intelligence Review, 52(2), 1089-1106.

[14] Ghosh, S., Das, N., Das, I., & Maulik, U. (2019). Understanding deep learning techniques for image segmentation. ACM Computing Surveys (CSUR), 52(4), 1-35.

[15] Sparavigna, A.C. (2016). A method for the segmentation of images based on thresholding and applied to vesicular textures. PHILICA Article number 889. Available preprint arXiv:1612.01131, 2016 - arxiv.org

[16] Sparavigna, A. C. (2017). Measuring the blood cells by means of an image segmentation. PHILICA Article number 1176. Available hal-01654006 hal.archives-ouvertes.fr

[17] Amelia Carolina Sparavigna, Roberto Pisano, Antonello A. Barresi. Measuring the size of pores by the segmentation of images from scanning electron microscopy. 6th European Drying Conference (EuroDrying 2017), Jun 2017, Liège, Belgium. Available hal-01574643, https://hal.archives-ouvertes.fr/hal-01574643/

[18] Arsiccio, A., Sparavigna, A. C., Pisano, R., & Barresi, A. A. (2019). Measuring and predicting pore size distribution of freeze-dried solutions. Drying Technology, 37(4), 435-447. Published online: 16 Feb 2018

[19] Sparavigna, A. C. (2017). Image Segmentation Applied to the Study of Micrographs of Cellular Solids. International Journal of Sciences, 6(02), 68-76. DOI: 10.18483/ijSci.1201

[20] Sparavigna, A. C. (2016). Analysis of a natural honeycomb by means of an image segmentation. PHILICA, Article number 897. 14 December 2016. Available ⟨hal-01416832⟩ hal.archives-ouvertes.fr

[21] Sparavigna, A. C. (2017). Image segmentation applied to micrographs of microcellular plastics. Philica, Article n. 953, 5 February 2017. Available ⟨hal-01456692⟩ hal.archives-ouvertes.fr

[22] Sparavigna, A. C. (2017). Measuring the particles in fly ash by means of an image segmentation. PHILICA. Article number 1105. Published on 30th August, 2017. Available <hal-01579285> hal.archives-ouvertes.fr

[23] Sparavigna, A. C. (2017). Image Segmentation Applied to the Analysis of Fabric Textures. PHILICA Article number 1157. Available hal-01633061 hal.archives-ouvertes.fr

[24] Sparavigna, A. C. (2017). An image segmentation for the measurement of microstructures in ductile cast iron. PHILICA Article number 1159. Published on 15th November, 2017. Available hal-01635665 hal.archives-ouvertes.fr

[25] Sparavigna, A. (2017). Measuring the size of tubules in phloem and xylem of plants. PHILICA Article number 1104. Available hal-01578826 hal.archives-ouvertes.fr

[26] Sparavigna, Amelia Carolina, Image Segmentation Applied to Satellite Imagery for Monitoring Water in Lakes and Reservoirs (January 11, 2018). PHILICA, Article n.1214. Available at SSRN: https://ssrn.com/abstract=3139574

[27] Marazzato, R., & Sparavigna, A. C. (2019). Using Clustering and Controlled Regression to Analyze the Orientation of Rod-Like Objects. International Journal of Sciences, Volume 8(05). Pages 16-18 DOI: 10.18483/ijSci.2055

[28] Sparavigna, A. C. (2019). A method based on image segmentation for the analysis of the orientation of rod-like objects. Zenodo. DOI: 10.5281/zenodo.2567394

[29] Arsiccio, A., Sparavigna, A. C., & Barresi, A. (2019). Image Segmentation Applied to the Investigation of Craquelure Domains in Paintings. EuroDrying'2019, Jul 2019, Torino, Italy. hal-02182582

[30] Abràmoff, M. D., Magalhães, P. J., & Ram, S. J. (2004). Image processing with ImageJ. Biophotonics international, 11(7), 36-42.

[31] Collins, T. J. (2007). ImageJ for microscopy. Biotechniques, 43(S1), S25-S30.

[32] Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. Nature methods, 9(7), 671.

[33] Ferreira, T., & Rasband, W. (2012). ImageJ user guide. ImageJ/Fiji, 1, 155-161.

[34] Girish, V., & Vijayalakshmi, A. (2004). Affordable image analysis using NIH Image/ImageJ. Indian journal of cancer, 41(1), 47.

[35] Schindelin, J., Rueden, C. T., Hiner, M. C., & Eliceiri, K. W. (2015). The ImageJ ecosystem: An open platform for biomedical image analysis. Molecular reproduction and development, 82(7-8), 518-529.

[36] Rueden, C. T., Schindelin, J., Hiner, M. C., DeZonia, B. E., Walter, A. E., Arena, E. T., & Eliceiri, K. W. (2017). ImageJ2: ImageJ for the next generation of scientific image data. BMC bioinformatics, 18(1), 529.

[37] Picht, E., Zima, A. V., Blatter, L. A., & Bers, D. M. (2007). SparkMaster: automated calcium spark analysis with ImageJ. American Journal of Physiology-Cell Physiology, 293(3), C1073-C1081.

[38] Jensen, E. C. (2013). Quantitative analysis of histological staining and fluorescence using ImageJ. The Anatomical Record, 296(3), 378-381.

[39] Hartig, S. M. (2013). Basic image analysis and manipulation in ImageJ. Current protocols in molecular biology, 102(1), 14-15.

[40] Schmid, B., Helfrich-Förster, C., & Yoshii, T. (2011). A new ImageJ plug-in "ActogramJ" for chronobiological analyses. Journal of biological rhythms, 26(5), 464-467.

[41] Papadopulos, F., Spinelli, M., Valente, S., Foroni, L., Orrico, C., Alviano, F., & Pasquinelli, G. (2007). Common tasks in microscopic and ultrastructural image analysis using ImageJ. Ultrastructural pathology, 31(6), 401-407.

[42] Ovesný, M., Křížek, P., Borkovec, J., Švindrych, Z., & Hagen, G. M. (2014). ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging.

Bioinformatics, 30(16), 2389-2390.

[43] Pool, M., Thiemann, J., Bar-Or, A., & Fournier, A. E. (2008). NeuriteTracer: a novel ImageJ plugin for automated quantification of neurite outgrowth. Journal of neuroscience methods, 168(1), 134-139.

[44] Igathinathane, C., Pordesimo, L. O., Columbus, E. P., Batchelor, W. D., & Methuku, S. R. (2008). Shape identification and particles size distribution from basic shape parameters using ImageJ. Computers and electronics in agriculture, 63(2), 168-182.

[45] Boudaoud, A., Burian, A., Borowska-Wykręt, D., Uyttewaal, M., Wrzalik, R., Kwiatkowska, D., & Hamant, O. (2014). FibrilTool, an ImageJ plug-in to quantify fibrillar structures in raw microscopy images. Nature protocols, 9(2), 457.

[46] Legland, D., Arganda-Carreras, I., & Andrey, P. (2016). MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. Bioinformatics, 32(22), 3532-3534.

[47] Grishagin, I. V. (2015). Automatic cell counting with ImageJ. Analytical biochemistry, 473, 63-65.

[48] Dong, Y., Pan, W. D., & Wu, D. (2019). Impact of Misclassification Rates on Compression Efficiency of Red Blood Cell Images of Malaria Infection Using Deep Learning. Entropy, 21(11), 1062.