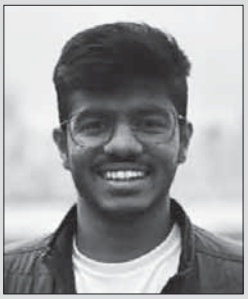
Image Analysis Methods for Grain Size Analysis : An Overview and A Case Study



HARSH VARDHAN GUPTA[1,2], GUAN-LIN CHEN[1], APURBA DAS[2], ASHOK. SHARMA[3], ANDREW CAMPBELL[4], PAUL MURRAY[4] AND NIKHIL GUPTA[1,*]

[1]Department of Mechanical and Aerospace Engineering, New York University, USA, [2]Department of Computer Science and Information Systems (CSIS), Secunderabad, [3]Ex-Head and Professor, Metallurgical and Materials Engineering Department, MNIT, Jaipur, [4]Department of Electronic and Electrical Engineering |University of Strathclyde, Glasgow, UK
* Corresponding author, **Email**: ngupta@nyu.edu

Abstract

The tools and techniques such as imaging and machine learning used in the measurement of many material and microstructural properties are rapidly evolving. In metals, the grain size is routinely measured to estimate the yield strength. This paper describes some of the algorithms used in processing the microstructures to conduct quantitative measurements. The image processing methods provide the possibility to go beyond calculating the ASTM grain size number and calculate the actual surface area of each grain, grain boundary length, and the shape of the grains. The image analysis methods can be very helpful in conducting detailed quantitative analysis with greater accuracy than many labour-intensive manual methods currently in use. The work describes the complexities in applying the imaging methods and approaches in the metallurgical and materials fields. Successful application of such methods can reduce the time and effort required to characterise microstructures and can provide more precise information.

Keywords: Microstructure, particles, image processing, machine learning.

Introduction

Grain size measurement is necessary for several reasons for ferrous and non-ferrous castings in the foundry industry such as microstructural : characterisation, material properties, process optimisation, quality control, performance prediction and failure analysis. Some common methods used to measure grain size in cast alloys are linear intercept method, planimetric method, and comparison with standard charts. Grain size can be estimated by visually comparing the microstructure of the cast alloy with standard grain size charts or reference images. This qualitative method provides a rough estimate of the grain size but is less accurate than quantitative techniques. The mechanical properties of metallic materials depend on the grain size. Hence, grain size measurement is a key task in the microstructure analysis of metals[1, 2]. American Society for Testing and Materials (ASTM) standard describes the Heyn Intercept method, which is widely used for grain size measurement. This method is labour-intensive and prone to human error and gives only an approximate measurement with large standard deviation.

With the advent of advanced imaging systems, the tools and techniques for material characterisation have undergone significant transformations[3]. Recent advancements in grain size measurement techniques have focused on enhancing accuracy, efficiency, and automation. Image processing and computer vision could also help significantly in reducing the time and manual inputs[4-6], while providing much greater details in the measurements such as the actual surface area of each grain, the length of grain boundaries, and morphological details about grain shapes[7]. Machine learning methods are immensely useful in microstructure recognition and deformation and failure analysis[8,9]. In machine learning, algorithms work with complex datasets to provide quantitative insight. They can learn patterns from large amounts of data and make predictions or decisions

based on those patterns. As some examples in the daily life, they are now recognising spam emails, predicting illnesses in a person before they occur, or even drive cars autonomously. Essentially, machine learning is a computer algorithm that learns from the available data to make predictions for a wide range of parameters. Since it is expensive and time consuming to conduct metallurgical analysis, automation using machine learning can help in reducing the time and cost and increasing the measurement accuracy.

Applying computer vision methods to an image dataset requires preparing the images to a standard format. The initial step is the of processing of micrograph into a digital format where grains are represented in terms of pixels[10, 11]. Once digitised, various algorithms can dissect these images, delineate grain boundaries, and quantify grain size. Image segmentation is a critical and essential component of many image analysis and/or pattern recognition systems, is one of the most difficult tasks in image processing, and determines the quality of the final result of analysis[12]. Segmentation can be performed either semantically, to label a set of pixels as belonging to a class (such as a material phase or boundaries), or on an instance basis, to label as set of pixels as belonging to an instance of a class (such as a single grain). The complexity of image segmentation stems from poor contrast that may result from insufficient etching, lighting or focus, overlapping features, clustering and precipitates, among others. The goal is to ensure that each segmented region retains homogeneity within a grain but the two neighbouring grains should exhibit a clear boundary between them.

In this work, a variety of algorithms are compared for their ability to quantitatively analyse micrographs. The methodologies adopted in this work offer an exhaustive analysis that surpasses the limitations of conventional ASTM grain size numbers. The surface area of each grain, grain boundary lengths, and the grain orientation are calculated. The refined post-imaging analytical techniques provide a richer, more detailed array of tools for analysing material properties.

Image processing for grain size measurement

The ASTM grain size number is measured as a whole number by visual comparison in the standard metallographic practice. However, modern processed specimens may have small differences in grain size, which may still be approximated by the same integer. Machine learning can help in more precise measurements of grain size. It can also detect presence of directionality or texture that the ASTM grain size analysis would not be able to capture. The process of preparing the micrographs and measuring the grain size is presented below.

Segmentation approach

The raw image (Fig 1 (a)) is taken from a Nikon Epiphot optical microscope and is read by using Open CV's library. The input dimensions of the raw image are (1920, 2560, 3). It is cropped to the dimensions of (1600, 2560, 3) to remove the scale bar before segmentation. The pixels to µm length ratio from the scale bar in raw image is 3.5 i.e., 350 pixels = 100 µm.

The primary step in the calculation of grain size is the detection of individual grain boundaries. Traditional methods for boundary detection such as Canny edge detection (CED)[13] have not provided satisfactory results. Canny edge detection is a classic edge detection algorithm developed by John Canny in 1986. It is commonly used for general-purpose edge detection tasks in computer vision applications. HED is a more advanced edge detection technique developed

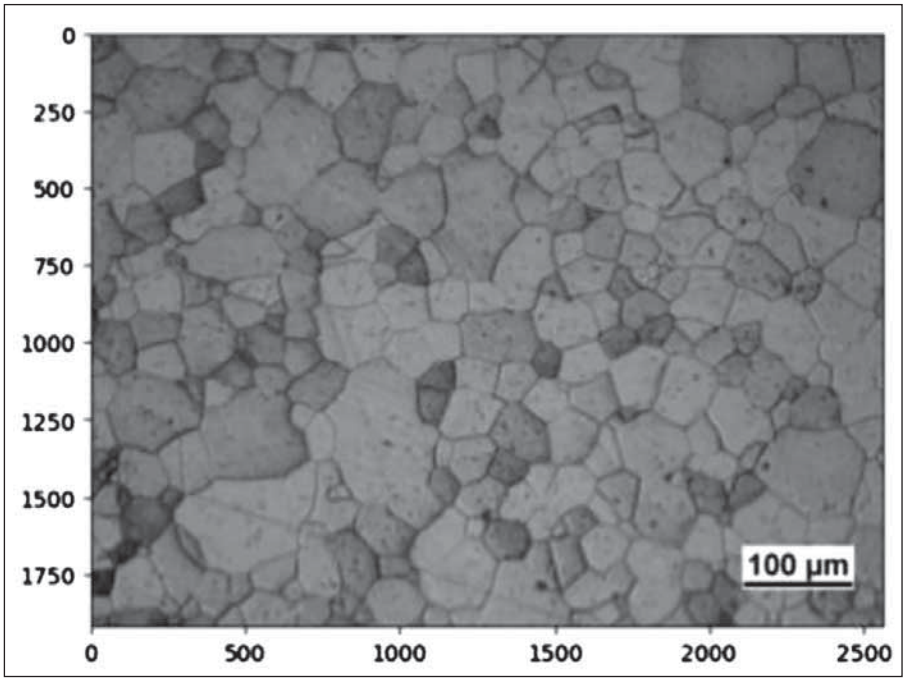


Fig 1: (a) Original microstructure of steel

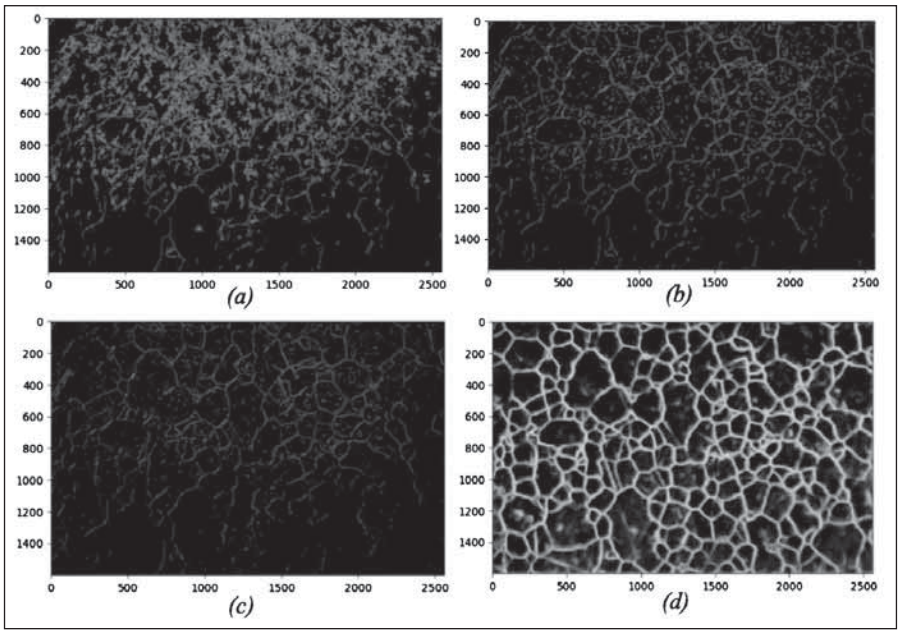


Fig 2: Illustration of grain boundary detection using CED and HED algorithms. (a), (b), (c) shows the CED by changing the lower (τ_l) and upper (τ_u) threshold value in hysteresis thresholding: (a) $\tau_l = 20$, $\tau_u = 150$, (b) $\tau_l = 70$, $\tau_u = 150$, (c) $\tau_l = 120$, $\tau_u = 150$. (d) HED shows a notable superiority over CED. The numbers on x and y-axes correspond to pixels.

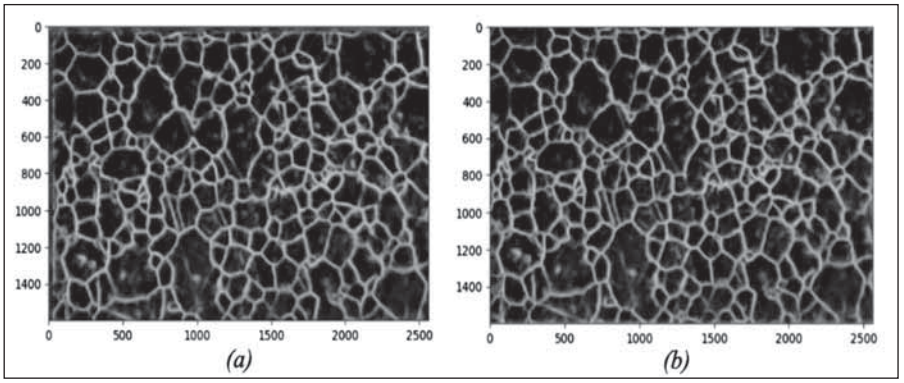


Fig 3: Demonstration of crop layer implementation in the original HED network on the input image: (a) without and (b) with crop Layer. The numbers on x and y-axes corresponds to pixels

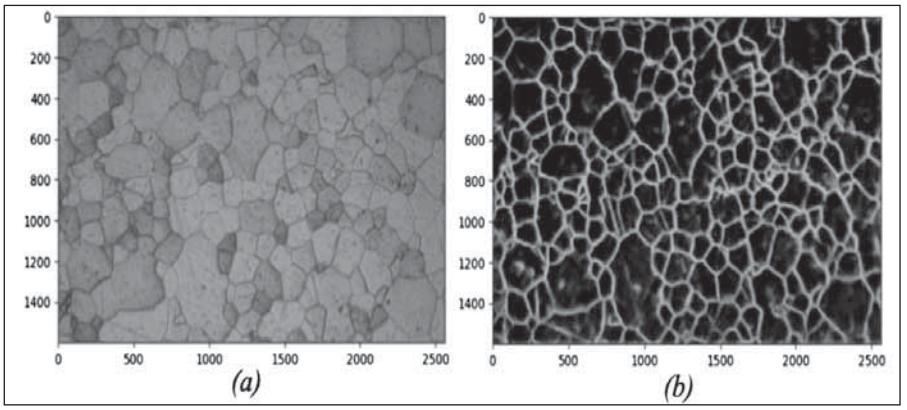


Fig 4: Illustration of the raw image that is fed to the neural network for obtaining the segmented grain boundaries: (a) raw and (b) output image after passing it into forward the HED network.

by Xie et al. in 2015. It aims to capture edges at multiple scales and levels of abstraction by combining features from multiple layers of a deep convolutional neural network (CNN). HED uses a deep neural network architecture to learn hierarchical representations of edges, which allows it to detect edges at different levels of details and complexity. Unlike Canny edge detection, HED is more tailored towards detecting complex and nuanced edges in images. While Canny edge detection is a traditional algorithmic approach, HED leverages the power of deep learning to achieve more sophisticated edge detection results. It's widely used for detecting a wide range of edges in images. Therefore, the technique that was utilised in this study is known as "Holistically Nested Edge Detection (HED)" [14].

Figure 2 compares the results obtained from the CED and the HED with different scale parameters.

Compared to the Canny algorithm, HED displays superior connectivity of edges, consistency and a spatial shift in the detection of edges[14]. This deep learning model employs fully convolutional neural networks for image-to-image prediction and directly produces the edge map image as an output[15, 16]. A crop layer is added to the network (Fig 3) as in the original network the input image gets shifted from the origin as demonstrated in Fig 3(b).

The raw image is processed prior to its transmission to the neural network to scale the image and subtract mean [17]. Scaling helps in reducing the computation time though it may decrease some features of the image. Based on a subjective evaluation of the images that yielded the best results, the scaling factor of 0.7 was selected. Figure 4(a) demonstrates

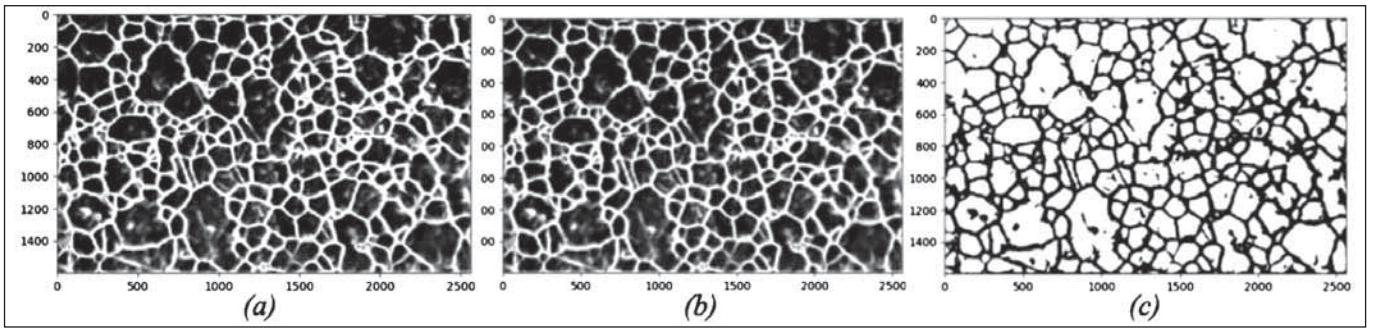


Fig 5: Illustration of postprocessing of the image for the labelling and extraction of individual grain properties: (a) contrast and brightness adjusted image, (b) gaussian blurred image, (c) inverted thresholded image. The numbers on x and y axes correspond pixels.

the raw image which is pre-processed using mean subtraction and scaling and then forward pass is performed to extract the edges. Figure 4(b) illustrates the edge map obtained from the HED network.

The contrast and brightness are now enhanced to detect the edges more efficiently. The first operation is to rescale the source image by a factor of λ and second is to offset by adding a δ term. The values of λ and δ are selected as 2 and 25, respectively, based on parametric studies to achieve the maximum grain boundaries detection efficiency. Figure 5(a) plots the image after contrast and brightness correction. Before extracting the individual grain boundaries, the algorithm performs Gaussian blurring on the adjusted image (Figure 5(b)) for smoothing of image, noise reduction and edge preservation. The blurred image is further thresholded using the combination of Binary Thresholding and Otsu's Binarisation and then inverted as shown in Fig 5(c). The connected components labelling on the thresholded image is performed to identify each individual grain so that measurements can be produced. This also helps in calculating the total number of grains and labelling them with different colours. The colours are randomly assigned to the grains in Fig 6.

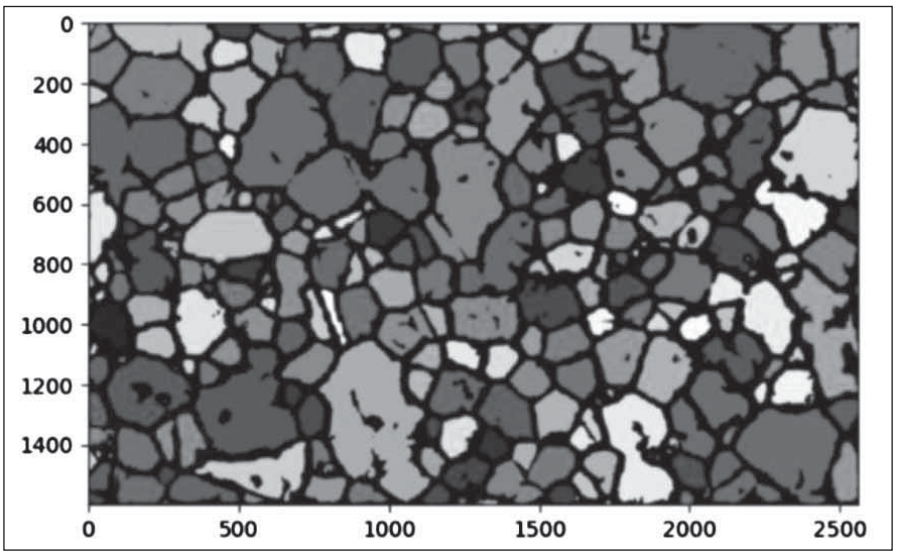


Fig 6: Grain Image. The numbers on x and y axes corresponds pixels

A number of techniques are available to measure the segmented objects. To extract the individual grain properties, the algorithm uses Skimage's measure module[18-20] with a Pandas data frame. The following properties are measured for each grain:

- **Area -** The number of pixels in the region, scaled by pixel-area. This area is then divided by $(pixel/\mu m)^2$ to scale the area in µm.
- **Equivalent Diameter** - The diameter of the grain with the same area as the region. This is divided by pixel_to_µm for scaling.
- **Perimeter -** Perimeter of the grain that uses a 4-connectivity to represent the contour as a line in the middle of the border pixels. This is also divided by pixel_to_µm for scaling.
- **Orientation** - Angle between the 0th axis (rows) and the major axis of the ellipse that has the same second moments as the region, ranging from -π/2 to π/2 counter-clockwise.
- **Major axis length or minor axis length** - The length of the major or minor axis of the ellipse that has the same normalised second central moments as the region. This is also divided by pixel_to_µm for scaling.
- **Centroid** - It returns the centroid coordinate tuple (row, column) of the grain.

In this image, the objects of equivalent diameter of fewer than 25 pixels are considered to be noise, and as a result, they are eliminated from the data frame. The border/edge touching grains can be discounted in the absence of complete information of particle to calculate the grain attributes.

$$G = 3.321928.\log_{10}n - 2.954 \quad \ldots\ldots \quad (1)$$

where n = number of grains per mm^2 area at 100 × magnification. To calculate the number of grains per mm^2 (n), the average grain area calculated above in μm^2 is multiplied by 10^6. Finally, the ASTM grain number (G) is calculated using Equation (1)[21].

Heyn intercept method

The intercept method involves plotting lines of known length on the micrograph and counting the number of times the line intercepts grain boundaries. The mean intercept length ($\bar{l}$) is given by[21]

$$\bar{l} = \frac{L_T}{P \cdot M} \quad \ldots\ldots \quad (2)$$

where L_T = total length of lines, P = total number of intercepts, and M = magnification. From $\bar{l}$, the ASTM grain size number G is calculated by [21]

$$G = -6.643856.\log_{10}\bar{l} - 3.288 \quad \ldots\ldots \quad (3)$$

This equation infers that the higher the grain size number, the smaller the average grain size.

The thresholded image from the HED algorithm (Fig 5(a)) is taken and noise reduction is performed by using a morphological operation called Opening, which is a combination of two techniques, i.e., erosion followed by dilation (Fig 7(a))[22]. Erosion erodes the foreground of an image depending on the size and shape of a structuring element. Dilation helps in connecting the broken parts of the object [22]. Morphological opening is effective for eliminating small structures from an image while retaining the shape and size of larger structures in the image. A structuring element with size of (10,10) and elliptical shape is used on the image to reduce the noise. To remove further noise, a median filter is then applied using a square kernel of size (5,5) as demonstrated in Fig 7(b).

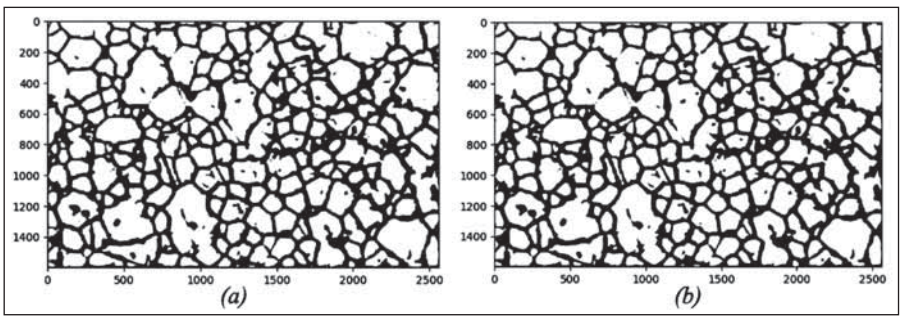


Fig 7: Noise removal on the input image after: (a) morphological transformation and then (b) median filtration. The numbers on x and y axes corresponds pixels

The denoised image is converted back to RGB image format, and 100 random vertical lines are drawn on the image with red colour (255, 0, 0) (as per the RGB convention). For the calculation of intersection points, the algorithm blends 2 images in α and (1- α) proportion as per

$$g(x) = \alpha.f_1(x) + (1 - \alpha).f_2(x) \quad \ldots\ldots \quad (4)$$

where $0 \leq \alpha \leq 1$ and $f_1(x)$ & $f_2(x)$ represents the images. The value of is chosen such that the resultant image has a particular range of mean pixel values when the vertical lines intersect the grain boundaries enabling their capture. Here, α = 0.4 and $f_1(x)$ being the image without the intercept line (Fig 8(a)) while $f_2(x)$ being the image which has the red intercept lines (as shown in Figure 8(b)). The intersection points of red and black pixels will have a mean range of colour channels between 140 and 160, which will assist in calculating the intersection points (shown in Fig 8(c)). The algorithm iterates over 100 vertical lines and counts intersection points and the mean intercept length ($\bar{l}$). Consequently, the grain size number is calculated using Equation (3) and the number of grains per mm^2 are calculated by Equation (1).

Results of grain size measurement

The two approaches ie, segmentation and intercept were executed on Jupyter notebook. Using the time library in python the wall-clock time was measured for both approaches. The segmentation approach found a total of 223 different grains and the output is presented in Table-1. This method yielded a grain size evaluation of 7.02 with an execution time of 15.01 seconds on Apple M1 Pro chip with 8 cores CPU and 14 cores GPU having 16GB of primary memory.

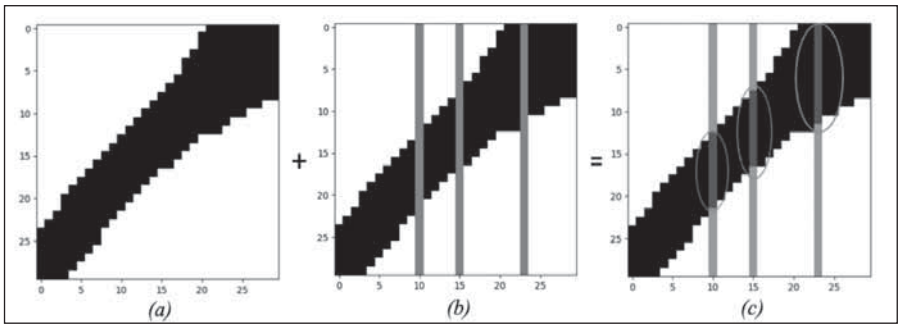


Fig 8: The detection of intersection points of grain boundary with the intercept lines. (a) The denoised image without the intercept lines having added weight (α = 0.4, (b) the denoised image with the intercept lines with having added weight (1- α = 0.6, (c) the oval regions indicate the change in color channel intensity for the intersection points. The numbers on x and y-axes correspond to pixels.

Table-1: Grain distribution data frame using segmentation approach. x_centroid and y_centroid corresponds to the x and y coordinate of the centroid of the particular grain respectively in pixels

No.	Area (µm)	Perimeter (µm)	Equivalent Diameter (µm)	Orientation (degrees)	Major Axis Length (µm)	Minor Axis Length (µm)	x_ centroid (pixels)	y_ centroid (pixels)
1	336.33	77.75	20.69	-38.03	27.69	16.45	42.66	37.22
2	2559.18	296.05	57.08	80.94	100.83	38.19	54.26	264.49
3	421.80	92.31	23.17	85.52	37.36	14.98	21.68	483.01
….	….	….	….	….	….	….	….	….
222	92.00	45.10	10.82	79.00	17.24	7.30	1582.11	543.97
223	56.57	43.75	8.48	88.75	19.19	4.07	1592.88	1328.58

A comparison of the results from the two approaches is shown in Table-2. Compared to the ASTM grain size number, which is a whole number, the grain size statistics computed using these image processing algorithms is more accurate and can differentiate between small differences in the grain size. The intercept approach was significantly faster than the segmentation approach.

Table-2: Results of segmentation vs intercept approach

Grain properties	Segmentation approach	Intercept approach
Average grain diameter (µm)	30.29 ± 18.61	43.01
Average grain area (µm^2)	991.37 ± 1422.05 (505.71 median)	688.38
Number of grains per mm^2	1008.71	1452.7
Execution time (s)	15.01	1.99
Grain size number*	7.02	6.46

*Unlike an integer ASTM grain size number, these numbers can be more precise.

Conclusions

The conventional methods for determining grain size, such as the Heyn Intercept method, are arduous and provide limited information. Image processing and computer vision have emerged as essential technologies, considerably cutting down the amount of time and effort needed to determine the grain size. Not only do these technologies offer a more effective method of measuring, but they also offer a more in-depth look into the microstructures and can help in differentiating small differences between two microstructures. The two approaches explained in this paper namely the segmentation and intercept approach help in capturing the grain properties like area, diameter and orientation, making it possible to have an in-depth understanding of the microstructure. The intercept approach in particular provides the grain size number in approximately 2 seconds which is significantly faster than the segmentation approach. In summary, while both approaches contribute to capturing grain properties in the microstructure of a casting, the segmentation approach offers a more detailed and nuanced analysis, allowing for a deeper understanding of spatial distribution and variability. On the other hand, the intercept approach provides a quicker overview, making it suitable for initial assessments or situations where time and resources are limited. The choice between the two approaches depends on the specific goals, requirements, and constraints of the analysis.

Acknowledgment

The authors thank Dr Rakesh Behera to provide the microstructure images. This work is supported by the National Science Foundation grant CNS-2234973.

References

1. Ralston, K., N. Birbilis, and C. Davies, Revealing the relationship between grain size and corrosion rate of metals. Scripta Materialia, 2010. 63(12): p. 1201-1204.
2. Aung, N.N. and W. Zhou, Effect of grain size and twins on corrosion behaviour of AZ31B magnesium alloy. Corrosion Science, 2010. 52(2): p. 589-594.
3. Minaee, S., Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz,

and D. Terzopoulos, Image Segmentation Using Deep Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. 44(7): p. 3523-3542.

4. Srivastava, H., H. Pearce, G. Mac, and N. Gupta, Determination of Fiber Content in 3-D Printed Composite Parts Using Image Analysis. IEEE Embedded Systems Letters, 2022. 14(3): p. 115-118.
5. Chen, G.L. and N. Gupta, Image processing and machine learning methods applied to additive manufactured composites for defect detection and toolpath reconstruction, in Machine Learning Applied to Composite Materials. 2022, Springer p. 19-44.
6. Stockman, G. and L.G. Shapiro, Computer Vision. 2001: Prentice Hall PTR.
7. Petrou, M.M. and C. Petrou, Image processing: the fundamentals. 2010: John Wiley & Sons.
8. Chowdhury, A., E. Kautz, B. Yener, and D. Lewis, Image driven machine learning methods for microstructure recognition. Computational Materials Science, 2016. 123: p. 176-187.
9. Cui, J., G. Zeng, N. Gupta, et al., Microstructure and Tensile Property of Al_3Zr/Al-Cu-Ni-V Composite Prepared by In Situ Reaction. Journal of Materials Engineering and Performance, 2023.
10. Morel, J.-M. and S. Solimini, Variational methods in image segmentation: with seven image processing experiments. Vol. 14. 2012: Springer Science & Business Media.
11. Caminero, M.A., M. Lopez-Pedrosa, C. Pinna, and C. Soutis, Damage Assessment of Composite Structures Using Digital Image Correlation. Applied Composite Materials, 2014. 21(1): p. 91-106.
12. Cheng, H.D., X.H. Jiang, Y. Sun, and J. Wang, Color image segmentation: advances and prospects. Pattern Recognition, 2001. 34(12): p. 2259-2281.
13. Canny, J. F., A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. PAMI-8: p. 679-698.
14. Xie, S. and Z. Tu, Holistically-nested edge detection, in Proceedings of the IEEE international conference on computer vision. 2015. p. 1395-1403.
15. Long, J., E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
16. Lee, C.-Y., S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, Deeply-Supervised Nets, in Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, L. Guy and S.V.N. Vishwanathan, Editors. 2015, PMLR: Proceedings of Machine Learning Research. p. 562-570.
17. Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning. 2016, MIT Press.
18. Garbe, C. and B. Ommer, Parameter Estimation in Image Processing and Computer Vision. 2013.
19. Burger, W. and M. Burge. Principles of Digital Image Processing - Core Algorithms. in Undergraduate Topics in Computer Science, 2009.
20. Reiss, T.H. Recognizing Planar Objects Using Invariant Image Features. in Lecture Notes in Computer Science. 1993.
21. ASTM International, Standard Test Methods for Determining Average Grain Size ASTM E1 12-13, 2021.
22. Beyerer, J., F. Puente León, and C. Frese, Morphological Image Processing, in Machine Vision: Automated Visual Inspection: Theory, Practice and Applications, J. Beyerer, F. Puente León, and C. Frese, Editors. 2016, Springer Berlin Heidelberg: Berlin, Heidelberg, p. 607-647.