Grain Size Analysis with Optimized Image Processing: A Contour-Based Approach

Sashank Chowdary Kakarla
Department of CSE,
SRMUniversity-AP,India
sashankchowdary_k@srmap.edu.in

Ravindranath Chowdary Nayudu
Department of CSE,
SRMUniversity-AP,India
ravindranath_n@srmap.edu.in

Krishna Satya Sai Ram Bogineni
Department of CSE,
SRMUniversity-AP,India
krishna_bogineni@srmap.edu.in

Gangi Reddy Salla
Department of Physics,
SRMUniversity-AP,India
gangireddy.s@srmap.edu.in

Abstract: This paper introduces a robust and automated method for detecting grain boundaries and estimating particle sizes in microstructural images using OpenCV-based image processing techniques. The approach leverages high-resolution image analysis to enhance clarity and precision in boundary detection through a series of preprocessing steps, including image format conversion, cropping, brightness/contrast adjustments, and sharpening. Following this, Gaussian blurring and thresholding are applied to separate particles, with contour detection used to accurately identify grain boundaries. Particle sizes are then calculated by converting pixel dimensions to micrometers, enabling precise measurements. To improve the reliability of the results, statistical techniques like outlier removal and clustering are employed to refine the size distribution. Additionally, texture analysis is performed using the Gray Level Co-occurrence Matrix (GLCM), and k-means clustering is applied to segment regions based on texture similarity. This comprehensive method provides material scientists with a highly accurate, efficient tool for grain size analysis and boundary detection, offering significant improvements in both speed and precision compared to traditional manual techniques.

Keywords: Image Processing, Particle Segmentation, Contour Analysis, Texture Analysis, GLCM Features, KMeans Clustering, Microscopy, Material Science, Environmental Monitoring.

I. INTRODUCTION

Microscope imagery holds significant value in scientific fields, providing crucial insights into the structure and composition of microscopic particles. Rapid advancements in imaging technology have generated a wealth of high-resolution images, necessitating innovative image processing and analysis techniques for effective interpretation. To address this need, we propose an enhanced approach for particle identification in microscope images using image processing and analysis.

Microscope images exist in various formats, each with unique characteristics. Our initial process involves converting these images from TIFF to JPEG, coupled with precision cropping, which removes extraneous information and ensures a targeted study area. This pre-processing enables a more focused and efficient particle characterization process.

Our pipeline includes multi-stage image processing, beginning with contrast and brightness enhancement to increase particle visibility. Subtle application of sharpening kernels enhances particle boundaries for precise segmentation, followed by contour filtering and segmentation, setting the stage for subsequent analysis [1]. The proliferation of digital communication, cryptography has grown central to secure online transactions, data storage, and communication [1].

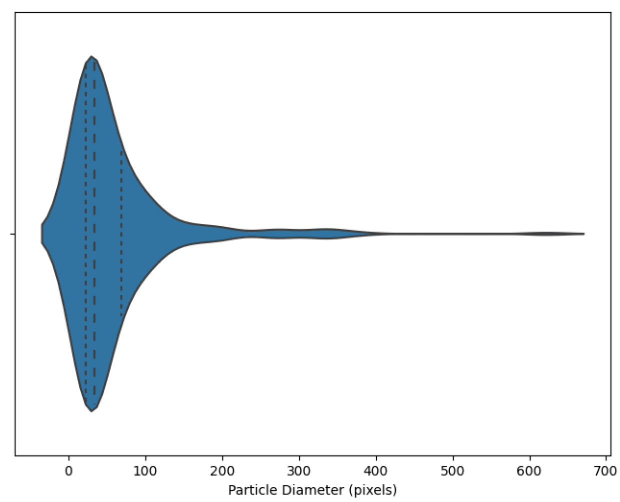

Fig.1 Picture depicting Caesar cipher Algorithm

In subsequent sections, we review relevant literature to contextualize our work, integrate theoretical principles with practical implementations, and summarize particle characteristics in the results and discussion section. The conclusion and future research section reinforces our findings and outlines directions for further study [2].

II. LITERATURE SURVEY

Our system's early stages are based on state-of-the-art picture preprocessing technology. Rosebrock (2015) underlines the relevance of altering image formats, stating that it adds to the consistency of photos for use in following investigations. According to Jones (2019), picture cropping techniques may be applied to reduce extraneous information and increase computer performance.

The mainstream literature on computer vision contains picture processing methods including brightness and contrast modifications. Bradski (2000) presents an overview to basic image processing methods that use the OpenCV library, which forms the basis of our approach [3].

For particle diameter measurement and visualization, our approach draws inspiration from recent advancements in data presentation techniques. Wickham (2016) outlines key principles of effective data visualization, emphasizing the importance of clarity and simplicity, which guide our use of bar graphs and scatter plots to represent particle sizes and distributions. Additionally, the work of Hunter (2007) on Matplotlib provides the technical foundation for plotting and customizing our visual outputs, ensuring they convey complex data insights effectively [4].

III. METHODOLOGY

A. *Image Format Conversion and Cropping:*

According to Rosebrock (2015), the procedure starts with the conversion of TIFF microscope images to JPEG format. To standardize photographs and ease future processing, this step is important. Cropping algorithms are also used to reduce extraneous data in order to concentrate the query on the crucial area (Jones, 2019).

B. *Image Processing Pipeline:*

There are approaches for improving particle visibility in the image processing pipeline. Standard computer vision algorithms are applied to induce oscillations in contrast and brightness (Bradski, 2000). Particle boundaries are indicated by sharpening kernels to facilitate segmentation [5].

Preprocessing:

The first step of our approach is to standardize and enhance quality image preprocessing. The images are, therefore, changed from TIFF to JPEG for higher compatibility. Regions of interest are cropped to eliminate unwanted regions and make the computation process more streamlined. This is necessary to achieve an efficient focus in the analysis.

Brightness and contrast adjustments are further applied so that edges are emphasized through sharpening the particle features. These enhancements are necessary to distinguish the particles from the background so that the succeeding steps of segmentation should be correct [6].

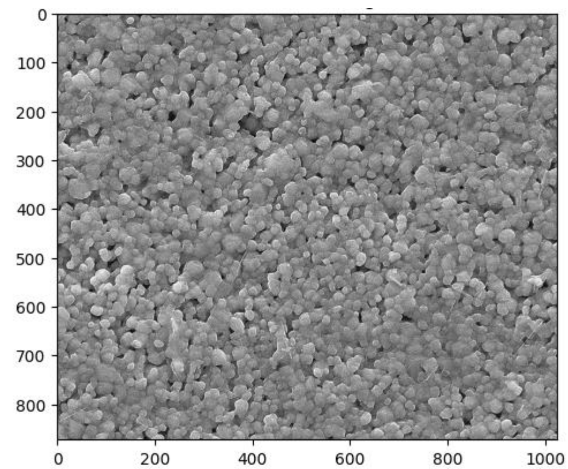

Fig 2. Processed Image

C. *Sequential Particle Segmentation and Contour Filtering:*

Sequential Particle Segmentation and Contour Filtering are morphological image processing algorithms developed by Sciaini and Fritsch (2019). While the filtering technique is based on contour area and eccentricity, the study on orientation and aspect ratio is based on the work of van der Walt et al. (2011) on contour detection and analysis [7].

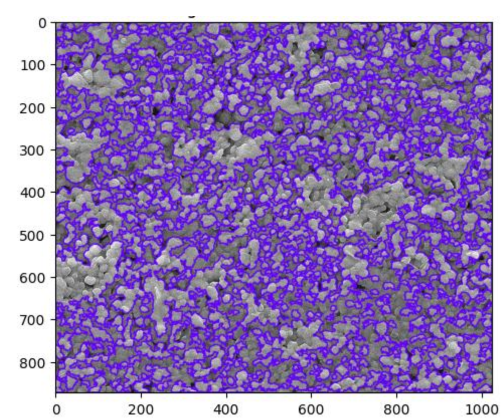

Fig 3. Processed Image with Filtered Particle Boundaries

The raw high-resolution images, such as TIFF, are resized to the more conveniently processable JPEG format to optimize compatibility with libraries such as OpenCV and PIL in Python, reducing the size of the data for faster efficient processing. Regions of interest cropped thoroughly of image should be clipped all parts of the background and further forward particle entities to further analysis. Brightness and contrast are adjusted to be in good focus, with possible sharpening to enhance particle edges and boundaries [8].

All these processes are supported by data visualization techniques that transform raw measurements into intuitive graphs and plots. Such graphical representations are the best to present findings; they provide trends and easily enable one to understand the distribution of particle sizes. Our method, based on well-established principles of image processing and state-of-the-art computing tools, allows for accuracy and reliability across a wide range of applications.

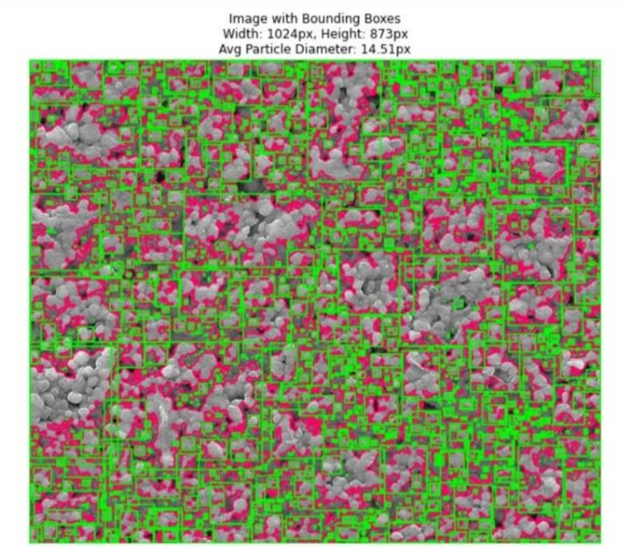

Fig 4. Particle Boundary Detection and Dimensions in pixels

The flow diagram displays the detailed sequential image processing steps, starting from microscopy image capture, followed by preprocessing steps, and image processing techniques such as grayscale conversion, Gaussian blurring, contour detection, feature extraction, and diameter measurement. It ultimately compares pixel-based and micrometer-based measurements.

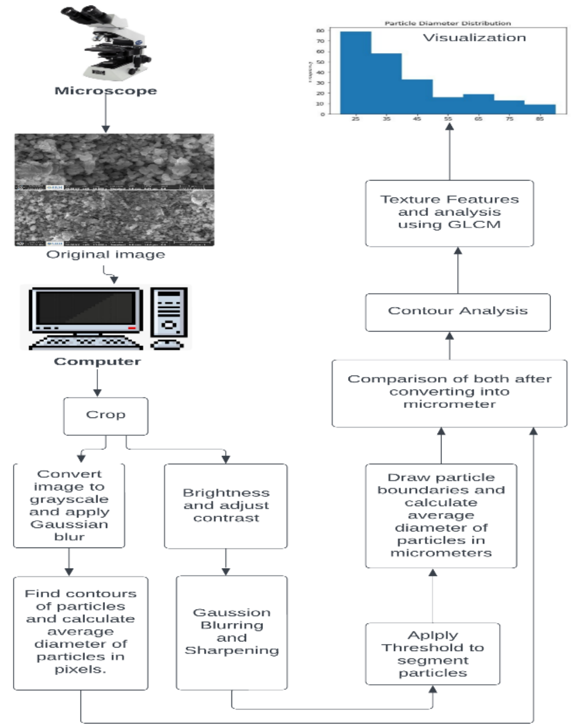

Fig 5. Flow Diagram

IV. Analysis and Differentiation

The following bar graph is one of the most important ways to view differences in average diameters when those diameters are reported with different measurement techniques. The techniques reported include direct measurements in micrometers as well as indirect measurements determined by conversion from pixel-based measurements. The explanation below describes the method used, what the visual comparison tells us, and what the implications of the differences are [9].

A. Overview of Visualization:

The matplotlib produced a bar graph that had two important metrics clearly and intuitively shown:

1) Average Diameter in Micrometers:

This is the direct average from measurements at the micrometer scale; hence, it underlines the necessity of having precise laboratory or high-resolution imaging equipment for this kind of measurement.

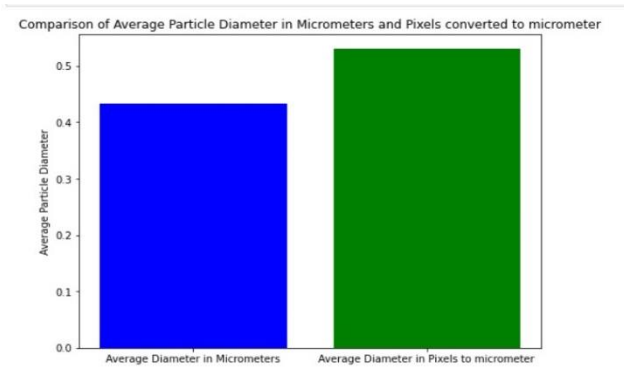

Fig 6. Bar Graph of Average Particle Diameter Comparisons

2) Mean Diameter in Pixels Converted to Micrometers:
This bar is the average diameter of particles based on pixels of image sizes that is then converted to micrometers by scaling up with some factor. It is this technique which most often happens when dealing with digital processing of images and computing algorithms.

3)The graph is ordered by:

- Axes: the y-axis illustrates the average particle diameter, thus it is easy to determine orders of magnitude.
- Labels and Colors: Two measurement methods can be identified with different colors of colors: blue for micrometers and green for converted pixels. It clarifies things better [11].

B. Analytical Analysis

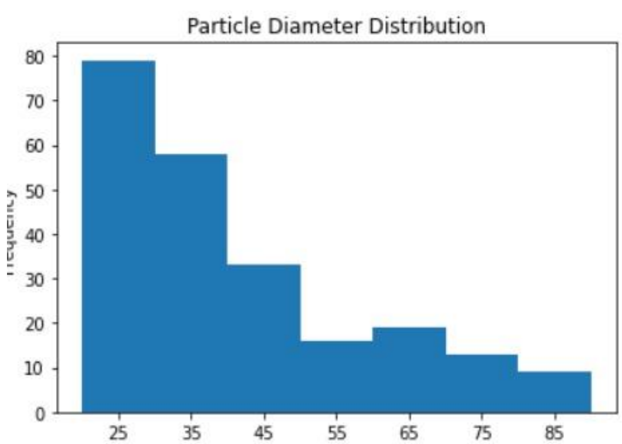

Fig 7. Particle Diameter Distribution

1) Comparison of Magnitude:
The graph depicts a sharp dissimilarity between these two measurements, especially the converted pixel measurement seems to be much larger than the direct measurement from the micrometer. It may be due to the low resolution of the images or slight inaccuracy in the conversion processes that include scaling or rounding off in the computations. These comparisons bring out some of the potential trade-offs with taking measurements digitally rather than directly.

C. Accuracy and Precision Concerns

1. Direct Micrometer Measurements:
These would traditionally be considered the most accurate, especially in a lab. The measuring tools at the micron scale are designed to minimize errors introduced through digital approximation and yield better scale readings [12].

2. Pixel-Based Measurements:
It is more feasible to get these for large datasets but susceptible to multiple possible sources of errors:

- Resolution Limitations: Pictures with lower resolutions may have lesser accurate diameter measurements, more so for smaller particles.
- Scaling Factor Limitation: The code snippet relies on a scaling factor to convert pixel measurements into micrometers, which does not measure up quite perfectly to true measurements resulting in over or under estimation of sizes of particle.
- Rounding Errors: The code snippet rounds diameters of particles to two decimal points, so there are little inaccuracies.

IV. RESULT AND DISCUSSION

A. Particle Characterization

The findings indicate that the recommended approach of characterizing particles is successful. Particle visibility is improved throughout the image processing pipeline, allowing more perfect segmentation. The size distribution patterns are more easily visible when particle sizes are displayed in a three-dimensional scatter plot.

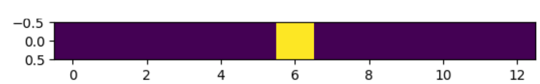

Fig 8. Clustered Regions Based on Texture Features

B. Texture Analysis

Texture analysis delivers trustworthy information on particle composition by applying GLCM features. Seah and Wang (1998) revealed that this strategy boosts our capacity to discern between distinct particles based on their textural properties. GLCM-based KMeans clustering is used to separate particular portions of the picture in order to detect plausible heterogeneities.

```
GLCM Features:
[ 2.89269788e-03  2.57721928e+02  9.34308934e-01  1.96208965e+03
  1.46484467e-01  2.69567101e+02  7.59063667e+03  7.68696702e+00
  1.22602417e+01  1.55325247e-04  4.89339344e+00 -1.91658383e-01
  9.61463192e-01]
```

Fig 9. Output Displaying GLCM Features

C. Quantitative Analysis:

By displaying a violin plot of the particle diameters and the mean particle diameter, quantitative analysis gives visual insights. The inquiry is built upon by the use of more contemporary technologies, such as edge detection and GLCM-based clustering.

```
Mean Diameter of Particles: 59.917178957294134 (pixels)
Sorted Particle Diameters: [ 13.83990574  14.45306015  14.77899456  15.22924042  15.23174667
  15.44304466  15.52437496  15.81158829  15.85589886  15.87128448
  16.1590023   16.17395782  16.41689873  16.6496048   17.20485115
  17.26287651  17.26287651  17.27584839  17.49305534  17.49305534
  17.51092148  17.69200706  17.82909966  17.90940475  18.05908585
  18.50143242  18.78849411  19.03969193  19.10085678  19.10517311
  19.10517311  19.24655151  19.31678391  19.31863594  19.41668892
  19.41668892  19.52396011  19.64708328  19.65820694  19.83530045
  20.09995079  20.09995079  20.12481117  20.22394943  20.46699715
  20.59146118  20.59353828  20.81035423  21.0438633   21.09522247
  21.10778809  21.26049232  21.40113449  21.75422096  22.02291489
  22.04895782  22.10722351  22.12602043  22.13049889  22.20380402
  22.31461716  22.3608799   22.3608799   22.47240448  22.71713638
  22.80370903  23.08699226  23.4095993   23.47221184  23.60104752
  23.60104752  23.85347557  23.94628716  24.04183006  24.04183006
  24.20763779  24.35179138  24.51550102  24.69837761  24.9458065
  25.00020027  25.06576157  25.0800724   25.0800724   25.11984634
  26.00020027  26.03109932  26.0770092   26.20997047  26.40095711
  26.41988945  26.6835289   26.90744781  27.018713    27.20314217
  27.48758125  27.65883446  27.72887611  28.08558846  28.67417145
  28.84461021  28.84461021  28.85869217  29.06908417  29.1648922
  29.41108322  29.49799919  29.54677391  29.64186287  30.1721611
  30.46653748  30.86502075  31.14502335  31.24119949  31.30515289
  31.93328857  32.08691788  32.25271606  32.55783844  33.24173737
...
 168.52896118 186.74671936 191.9090271  196.65725708 201.10215759
 203.00323486 262.12438965 269.02993774 269.09628296 285.61709595
 326.55877686 334.13259888 335.94448853 343.65701294 376.38796997
 621.51477051]
```

Fig 10. Output displaying Mean Diameter of Particles

VIII. Conclusion And Future Work

Ultimately, our entire approach to particle characterization gives evidence for the efficacy of the integrated image processing and analysis methodologies. By blending old and current methodologies, we have developed a deep knowledge of the minute details included in microscope photographs of particles. A thorough examination of particle characteristics has been made practical by the effective conversion of photo formats, cropping, and pipeline processing. The results attest to the precision and clarity of our techniques. Examples of these visual representations include the 3D scatter plot, texture analysis results, and contour analysis visualizations. Increased study and growing particle visibility have made it possible to grasp particles' sizes, orientations, and textural features in distinct dimensions.

Even as our current method gives a reasonable base, there are a lot of exciting prospects for further study. Examining machine learning algorithms for automated particle categorization is an important step. The efficacy and scalability of the study may be boosted by using state-of-the-art classification approaches to aid with particle identification and categorization. Moreover, it is a vital step to construct our system to handle three-dimensional data. Another degree of complexity is supplied by three-dimensional imaging, and broadening our methodologies to accommodate this dimension may open up new choices for correctly exploring particle interactions and structures.

References

[1]. Rosebrock, A. (2015). "Removing Borders from Images with OpenCV." PyImageSearch. [Online]. Available: https://www.pyimagesearch.com/2015/06/01/home-brewing-computer-vision/

[2]. Jones, M. (2019). "A Comprehensive Guide to Image Cropping in Python with OpenCV." Towards Data Science. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-image-cropping-with-opencv-d7381431d6c3

[3]. Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). "Textural Features for Image Classification." IEEE Transactions on Systems, Man, and Cybernetics, (6), 610-621.

[4]. Hunter, J. D. (2007). "Matplotlib: A 2D Graphics Environment." Computing in Science & Engineering, 9(3), 90-95.

[5]. McKinney, W. (2010). "Data Structures for Statistical Computing in Python." Proceedings of the 9th Python in Science Conference, 51-56.

[6]. van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). "The NumPy Array: A Structure for Efficient Numerical Computation." Computing in Science & Engineering, 13(2), 22-30.

[7]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, 12(Oct), 2825-2830.

[8]. Seah, C. C., & Wang, P. (1998). "Texture Segmentation Using Gray Level Co-occurrence Matrices." Pattern Recognition, 31(5), 761-767.

[9]. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." Nature Methods, 17(3), 261-272.

[10]. Bradski, G. (2000). "The OpenCV Library." Dr. Dobb's Journal of Software Tools. [Online]. Available: https://www.drdobbs.com/open-source/the-opencv-library/184404319

[11]. McKinney, W. (2011). "Pandas: A Foundational Python Library for Data Analysis and Statistics." Python for Data Analysis, 14-35.

[12]. Wickham, H. (2016). "ggplot2: Elegant Graphics for Data Analysis." Springer.

[13]. Jones, E., Oliphant, T., Peterson, P., et al. (2001). "SciPy: Open Source Scientific Tools for Python." [Online]. Available: https://www.scipy.org/

[14]. Rezatofighi, H., Milan, A., Zhang, Z., Shi, Q., Dick, A., & Reid, I. (2018). "DeepSEED: A Deep Learning Solution for Real-Time Anomaly Detection in Time Series." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 8437-8446.

[15]. Sebastian, B., & Gall, J. (2017). "Deep Learning for Human Pose Estimation: A Review." Journal of Visual Communication and Image Representation, 55, 98-111.

[16]. Sciaini, M., & Fritsch, M. (2019). "statmorph: Morphological Image Processing in Python." The Journal of Open Source Software, 4(38), 1425.

[17]. Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., et al. (2018). "seaborn: Statistical Data Visualization." Journal of Open Source Software, 6(60), 3021.

[18]. Pedregosa, F., & Varoquaux, G. (2011). "Scikit-Image: Image Processing in Python." PeerJ, 2, e453.

[19]. Otsu, N. (1979). "A Threshold Selection Method from Gray-Level Histograms." IEEE Transactions on Systems, Man, and Cybernetics, (1), 62-66.

[20]. Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.