Evaluation of mechanical properties of Direct-Reduced Iron (DRI) using SEM Backscattered-Electron Images

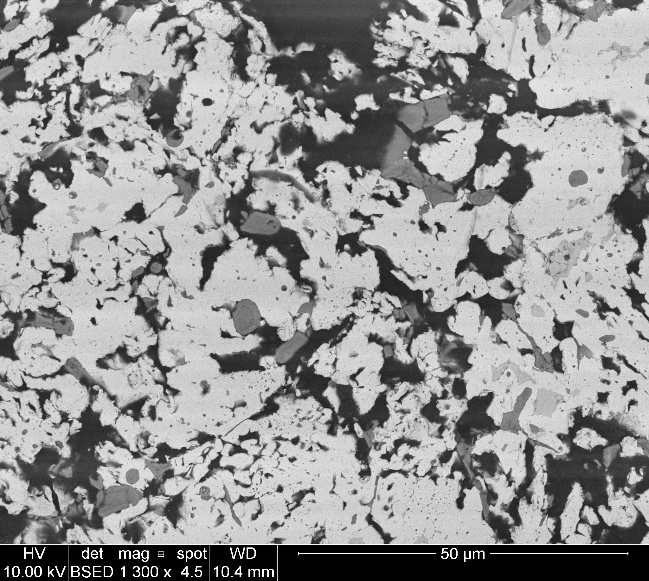
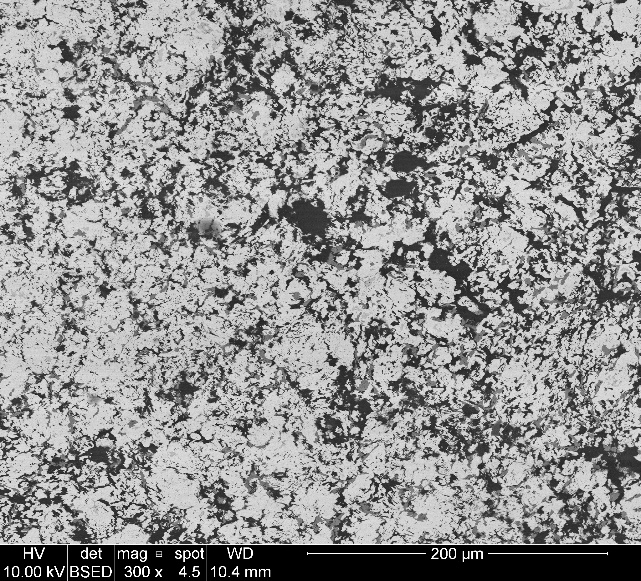
Name: Geonu Kim

E-mail: [rjsdnrlarla@gmail.com](mailto:rjsdnrlarla@gmail.com)

LinkedIn: [www.linkedin.com/in/geonu-kim-096060136](http://www.linkedin.com/in/geonu-kim-096060136)

# 1. Introduction

## 1-a. Problem Description



(a) 300 X

(b) 1,300 X

Figure 1. Scanning electron microscopy (SEM) backscattered electron images of cross-sections of resin vacuum-mounted commercial DRI samples. Cross-section at (a) 300x and at (b) 1,300x magnification. (Black is pore, white is reduced iron, light grey is unreduced iron, and dark grey is gangue.)

Direct-Reduced Iron (DRI), also known as sponge iron, is highly porous, much less dense than iron ore and sintered steel, and is used as a raw material in electric furnace steelmaking. DRI is expected to sustain compressive impacts during handling, stockpiling, and shipping, which tend to break down DRI into fines. Fines are undesirable, requiring special handling (briquetting, for example) and potentially causing loss of material. The loss of DRI mostly occurs by cracking; the compression strength measured according to ISO 4700 is generally used to benchmark the physical properties of industrial pellets.

The strength of DRI can be evaluated using the BSE image of DRI cross-section (Figure 1). However, processing the SEM backscattered electron image is time-consuming and not automated yet; DRI backscattered electron image analyzer, a new software proposed here, can improve the quality of the image and compute mechanical properties within a few seconds. A metallurgical scientist studying ironmaking and steelmaking, DRI plant operator, DRI technology specialist, DRI process designer and engineer are expected software users.

## 1-b. Objective for Projects

This project aims to create software for the estimation of the physical and mechanical properties of direct-reduced iron (DRI) from their SEM backscattered electron images. The purpose of the software is to process the SEM cross-section images of DRI to measure the portions of phases making up DRI samples, porosity, pore-size distribution, and all that sort of things which determine the physical and mechanical properties of DRI; based on the initial calculations, the software will finally compute the compressive strength of DRI.

## 1-c. Overview of Approaches Taken

The SEM Image of DRI is occasionally noisy because of experimental parameters including electron acceleration voltage, spot-size, beam focus, depth-of-focus, and astigmatism. Thus, it’s recommended for the software to have a built-in function which improves the selected BSE image quality. For this purpose, some core numeric libraries such as NumPy (Base N-dimensional array package), SciPy (Fundamental library for scientific computing), and Matplotlib (Comprehensive 2D plotting) were used for image processing and histogram generation. Useful methods in these scientific modules are only working on python 3, so python 3 is selected as a major interpreter. The fractions of phases covering the sample were determined based on the computed histogram of the image by using the software proposed here. The typical size of pore and pore distribution are also calculated to compute the strength of DRI. All the software calculations were designed based on general metallurgical science and powder metallurgy. For developing graphical user interface of the software, Tkinter is used to create simple GUI apps.

## 1-d. Organization of the rest of the report

1. Introduction 1

1-a. Problem Description 1

1-b. Objective for Projects 2

1-c. Overview of Approaches Taken 2

1-d. Organization of the rest of the report 2

2. Detailed features of the product design 3

3. Detailed features developed 4

4. Assessment of how well approach works for problem 8

5. Remarks 10

6. Appendix (Highlighted Codes) 11

2. Detailed features of the product design

* Improve the input SEM image qualities for better resolution and optimal brightness and contrast
* Process SEM backscattered electron images to discriminate the phases inside and measure the portions of these phases
* Process SEM backscattered electron images to calculate porosity, and pore-size distribution
* Identify the biggest crack and void to determine the critical stress of the sample
* Compute the physical properties of DRI based on the measured results above

# 3. Detailed features developed

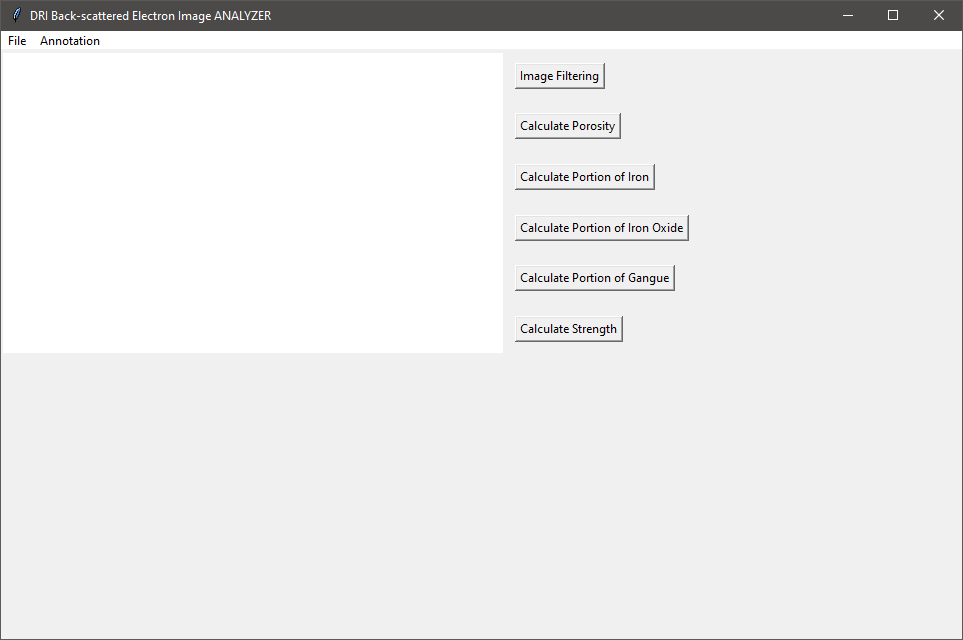


Figure 2. GUI of DRI Back-Scattered Electron Image Analyzer.

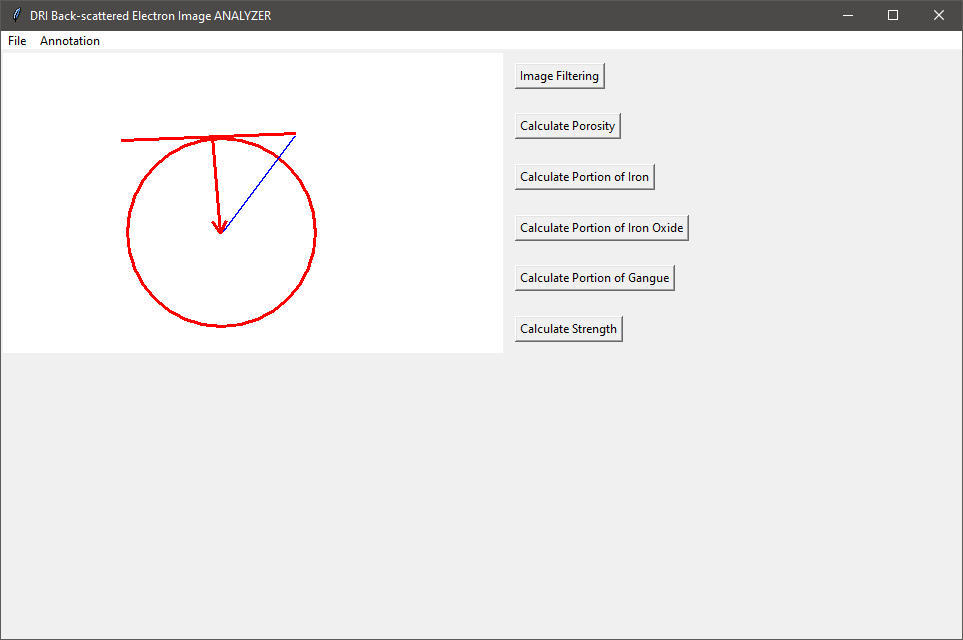


Figure 3. Drawing Canvas of DRI Back-scattered Electron Image Analyzer.

In this section, the main features of the software developed are explained. Figure 2 shows the GUI of the software when we first open it. In the menu bar, there is a File menu including new, open, save, save-as, exit, and about menu; and an annotation menu containing general canvas operations. Simple drawings can be drawn as shown in Figure 3.

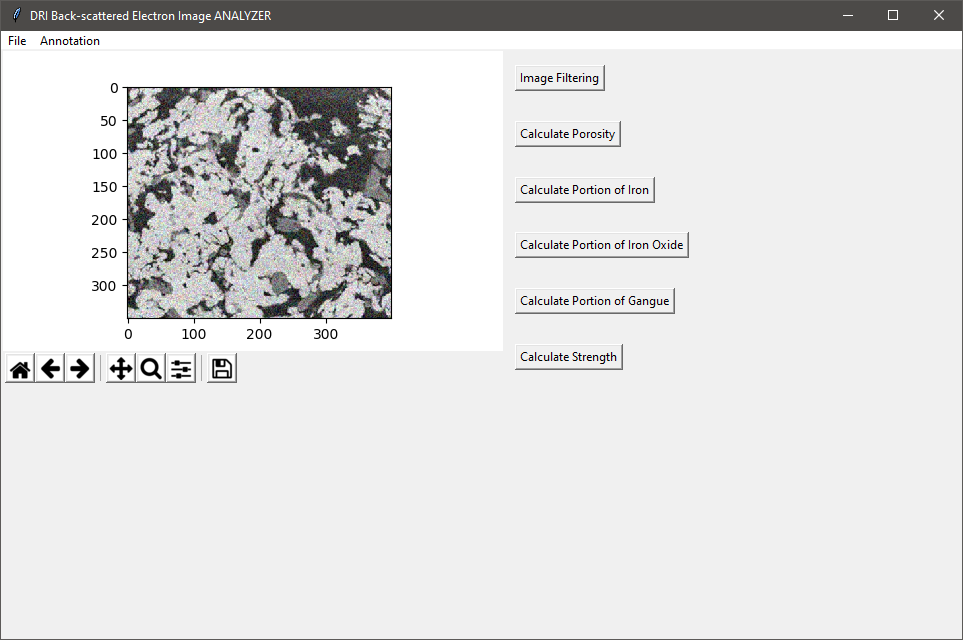


Figure 4. DRI SEM BSE Image loaded on the Analyzer.

Also, it is possible to open a BSE image as depicted in Figure 4; the image will be loaded on the upper-left section on the drawing canvas of GUI. A toolbar for simple image processing is located under the image loaded.

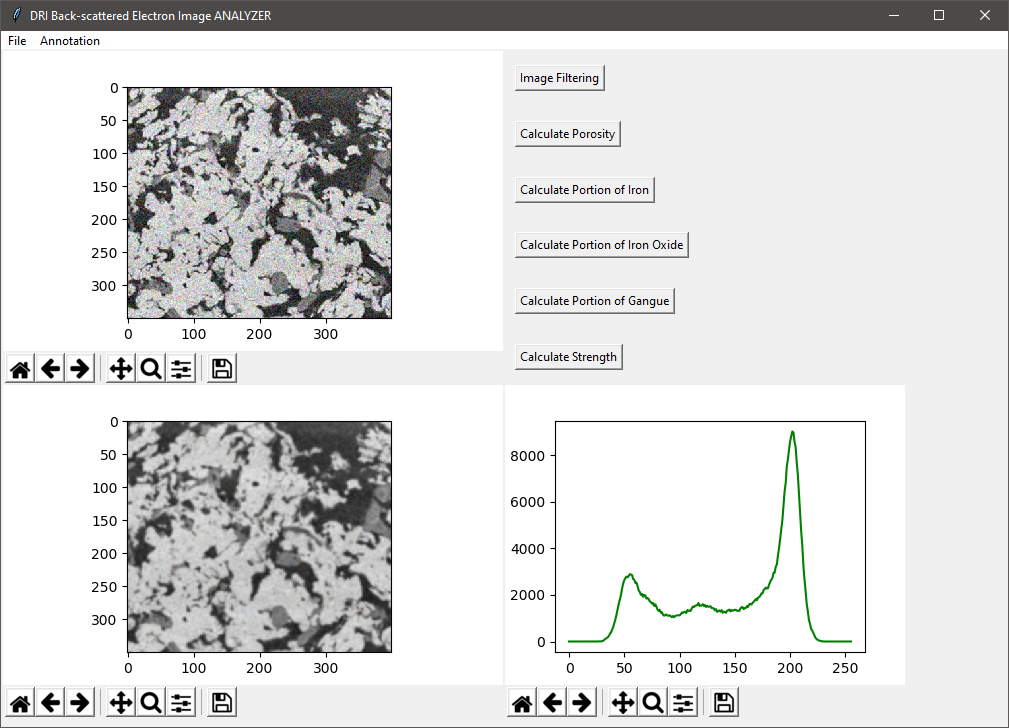
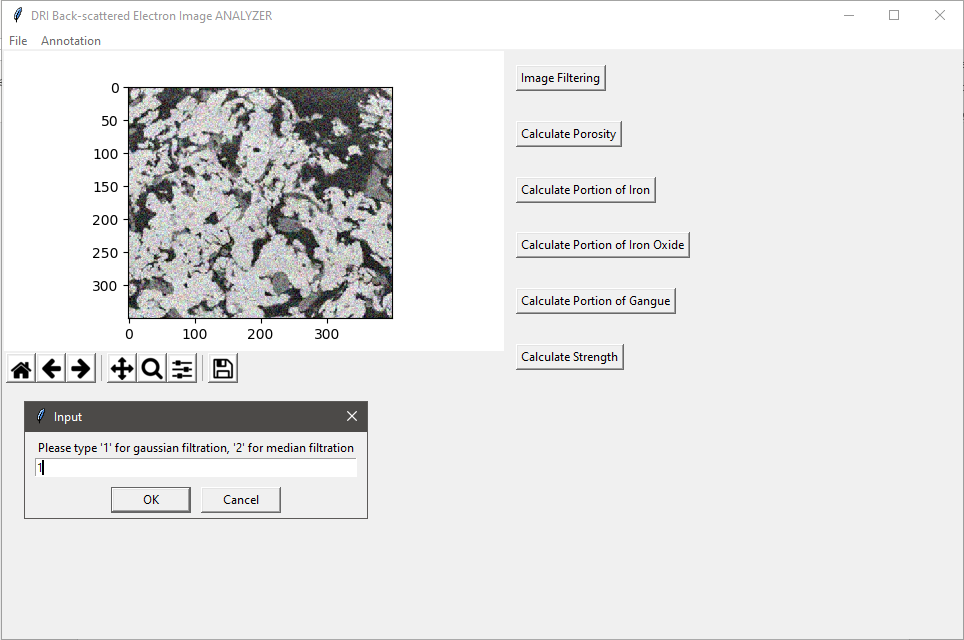


Figure 5. Filtration of SEM BSE Image with Gaussian or Median Filter

After loading an image, the image can be filtered by selected modes (Gaussian Filter or Median Filter). As a result, the software generated a filtered image on the bottom-left of GUI and a histogram plot computed based on the filtered image on the bottom-right of GUI (Figure 5).

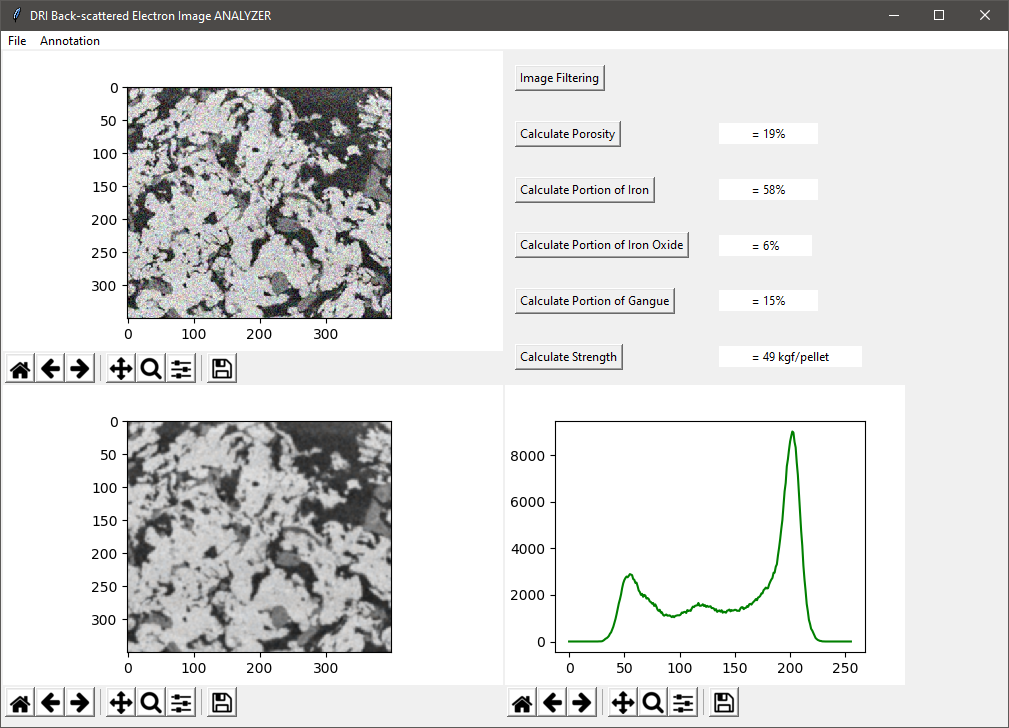


Figure 6. Computing Porosity, Phase Portions, and Mechanical Property of DRI

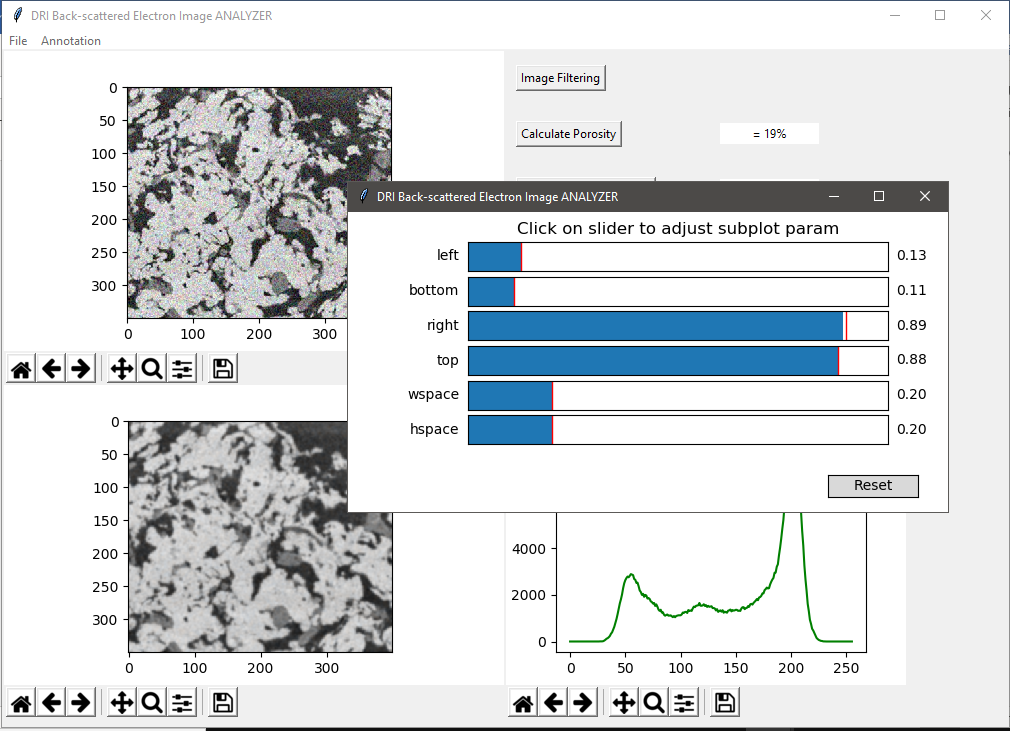
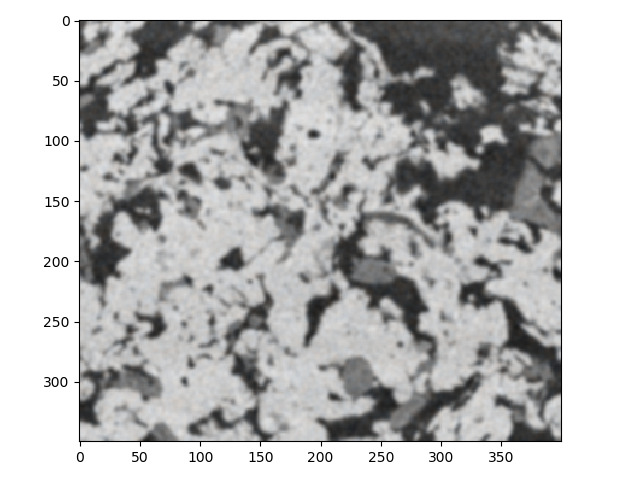
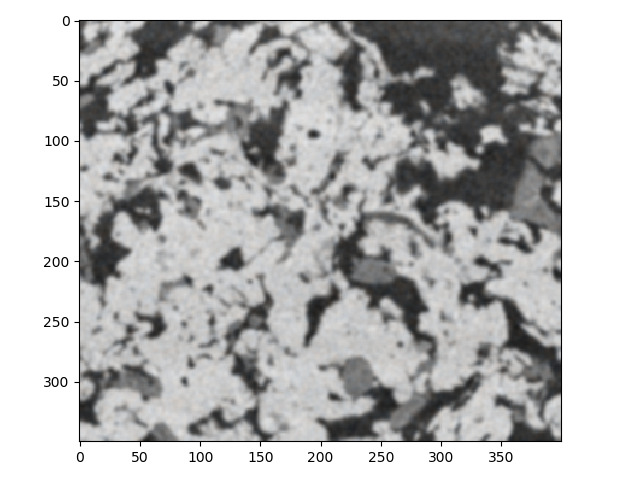
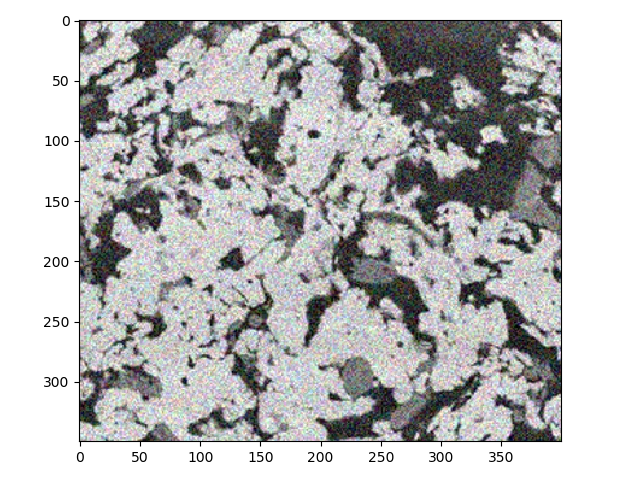


Figure 7. Modifying Images or Plot Parameters by Toolbar

Finally, users can calculate significant parameters for DRI mechanical properties within a second by clicking buttons on the upper-right side of GUI; DRI porosity, fractions of phases (iron, iron oxide, and gangue), and DRI strength (unit: kgf/pellet) can be computed quickly as indicated in Figure 6. Every image can be modified through an individual toolbar (Figure 7).

# 4. Assessment of how well approach works for problem



(a) Noisy raw BSE image of DRI cross-section

(b) Gaussian-filtered BSE image

(c) Median filtered BSE image

Gaussian filtering

Median filtering

Figure 8. Filtered images by applying Gaussian and Median filter for noise removal.

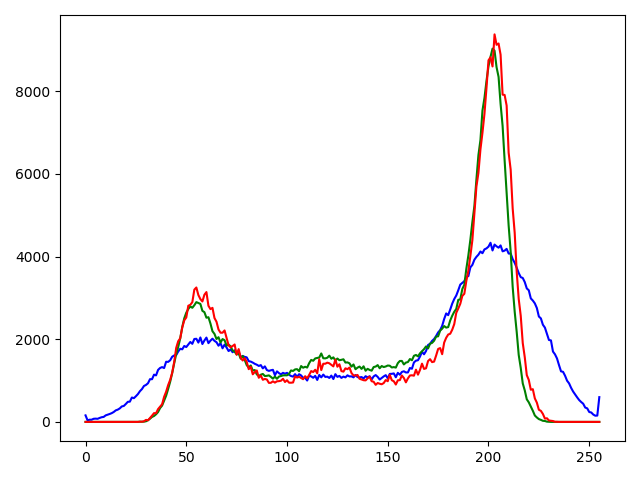
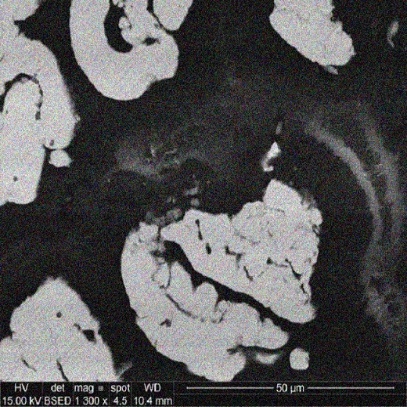
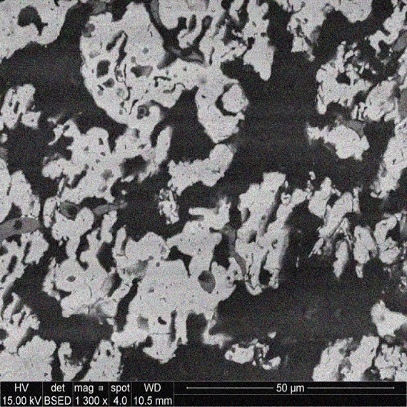
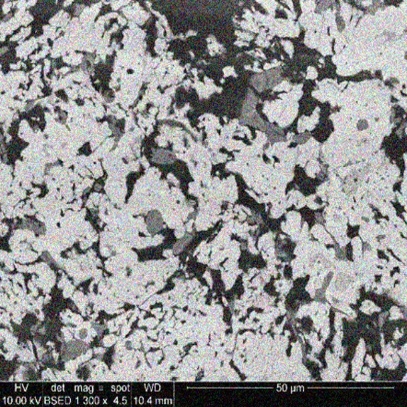


Figure 9. Comparison of the histogram of raw image and filtered images; Blue Line = Raw Image, Red Line= Median Filtered Image, Green Line = Gaussian Filtered Image.

Figure 8 shows the changes of the BSE images of DRI cross-section after applying the Gaussian filter and median filter to reduce noise level. The image filtering generates more refined images, which are more suitable to refine their histogram and better for scientific calculations. Figure 9 shows how the histogram changes depending on selected filters; more sharp peaks were detected in the filtered images (red and green line) compared with the histogram of raw image (blue line). As in Figure 8 and Figure 9, image filtration successfully improves the quality of images for phase discrimination.



(a) DRI-A

(b) DRI-B

(c) DRI-C

Figure 10. SEM backscattered electron images of cross-sections of resin vacuum-mounted DRI samples; (a) DRI-A, (b) DRI-B, and (c) DRI-C

Table 1. Comparison of Computation Result of DRI-A, DRI-B, and DRI-C



Figure 10 shows cross-sections of DRI-A, DRI-B, and DRI-C; DRI samples were produced in different conditions changing the inner structure of the samples. Compared DRI-C with the others, DRI-C had more open-structure with lots of pores (black area), indicating that DRI-C may have lower strength than that of the others because of biggest 3-dimensional defects in DRI. Table 1 summarizes the calculated results of DRI-A, DRI-B, and DRI-C by using the software. As expected, DRI-C had lowest strength (31 kgf/pellet) among the samples. The strength of DRI is strongly affected by the porosity of the sample. These calculated results matched with the common knowledge of powder metallurgy, proving that the software works well for the project problem.

# 5. Remarks

* To run the software, **python 3** interpreter must be used (not python 2).
* To run the software, the interpreter must include following libraries: matplotlib, numpy, scipy, PIL (Pillow).
* Software Running Instruction

Please run ‘App.py’ at the interpreter of python 3 including necessary modules.As a dry run, ‘DRI\_Image\_1.jpg’, ‘DRI\_Image\_2.jpg’, and ‘DRI\_Image\_3.jpg’ are attached. Please open these files to operate calculations. Open this file first using a file menu, then click the GUI buttons in following order: [image filtering], [calculate porosity], [calculate portion of iron], [calculate portion of iron oxide], [calculate portion of gangue], and [calculate strength].

If you have confusion, please refer to <https://www.youtube.com/watch?v=03C1qXk-r1I>; this video includes how to use the software.

# 6. Appendix (Highlighted Codes)

Highlighted code #1: Calculations.py (This code includes basic histogram interpretations)

|  |
| --- |
| **import** numpy **as** np  **def** DRIcalculations(Histogram):   *# print(np.sum(Histogram\_SEM\_Image1\_GaussianFilter[:200]) / np.sum(Histogram\_SEM\_Image1\_GaussianFilter))* Pores = Histogram <= 80  Gangues = np.logical\_and(Histogram > 80, Histogram <= 130)  IronOx = np.logical\_and(Histogram > 130, Histogram <= 150)  Iron = Histogram >= 150   porecount = np.sum(Histogram[:80]) / np.sum(Histogram)  ganguecount = np.sum(Histogram[80:130]) / np.sum(Histogram)  ironoxcount = np.sum(Histogram[130:150]) / np.sum(Histogram)  ironcount = np.sum(Histogram[150:]) / np.sum(Histogram)   *# ganguecount = np.count\_nonzero(Gangues)  # ironoxcount = np.count\_nonzero(IronOx)  # ironcount = np.count\_nonzero(Iron)* **return** porecount, ganguecount, ironoxcount, ironcount |

Highlighted code #2: ImageFiltering2.py (This code includes simple image filtration methods)

|  |
| --- |
| **from** scipy **import** ndimage  **def** GausianFiltration(ImageArray):  SEM\_Image\_GaussianFilter = ndimage.gaussian\_filter(ImageArray, sigma=1.5)  **return** SEM\_Image\_GaussianFilter  **def** MedianFiltration(ImageArray):  SEM\_Image\_MedianFilter = ndimage.median\_filter(ImageArray, size = 5)  **return** SEM\_Image\_MedianFilter |