SIEVE ANALYSIS OF ROCK FRAGMENTS USING ADVANCED IMAGE PROCESSING

Co-Author: Prof. R.K.Sinha

Author

Kritika Sinha

DISSERTATION

Submitted

To

CIL Innovation and Incubation Centre

Indian Institute of Technology

(Indian School of Mines)

India

DEPARTMENT OF MINING ENGINEERING
Indian Institute of Technology(Indian School of Mines),Dhanbad
Dhanbad, Jharkhand-826004

Date: 20-07-2024

CERTIFICATE

This id to certify that the Dissertation entitled **“SIEVE ANALYSIS OF ROCK FRAGMENTS USING ADVANCED IMAGE PROCESSING”** has been submitted by **Miss Kritika Sinha** of **Manipal Institute of Technology** (Admission Number: **220961164**) to **CIL innovation and Incubation centre, IIT(ISM)** in fulfilment of Summer Internship Project. Miss Kritika Sinha has worked under my Guidance and Supervision for the submission of this Dissertation, which to my Knowledge has reached the requisite standard.

Prof. R.K. SINHA
Head of
CIL, Innovation and Incubation Centre
Indian Institute of Technology
(Indian School of Mines)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my guide, **Professor R. K. Sinha**, Head of, Coal India Limited Innovation and Incubation centre, IIT(ISM), for his invaluable guidance, support, and encouragement throughout the course of this research work. His expertise and insights have been instrumental in shaping the direction of my dissertation on the topic of **"Sieve Analysis of Rock Fragments Using Advanced Image Processing Techniques."**

I am also deeply grateful to **Mr. Rajkishore Singh**, AGM Mining, Sonepur-Bazari Opencast Mine of Raniganj Area, ECL, for his immense support and assistance in facilitating the collection of rock fragment images from the mines. His cooperation and help have been crucial for the successful completion of this study.

My heartfelt thanks go to all the staff members of the Department of Mining Engineering, IIT(ISM), for their continuous support and assistance whenever needed. Their cooperation has been vital in the smooth progression of my research work.

Lastly, I take this opportunity to express my profound gratitude to my beloved parents for their untiring motivation, unwavering support, and constant encouragement, which have been a source of strength and inspiration throughout this journey.

Date: 20-07-24

Table of Contents

CERTIFICATE **3**

ACKNOWLEDGEMENT **5**

LIST OF FIGURES **11**

CHAPTER 1 .... **13**

INTRODUCTION **13**

- **1.1 Introduction to Mining Operations** 13
- **1.2 Unit operations in Mining Engineering** 14
- **1.3 Image analysis in Mining Operations** 17
- **1.4 Rock fragment Analysis** 19
- **1.5 Motivation for the Study** 20
- **1.6 Problem Statement** 20
- **1.7 Objectives of the Study** 20
- **1.8 Scope of the Study** 20
- **1.9 Methodology Overview** 20

CHAPTER 2 .... **21**

LITERATURE REVIEW **21**

- **2.1 Introduction** 21
- **2.2 History** 21
- **2.3 Innovative Advancements in Rock Fragmentation Analysis** 22
- **2.4 Key theories** 22
- **2.4 Conclusion** 23

CHAPTER 3 .... **25**

IMAGE PROCESSING **25**

- **3.1 Enhancement of images for contour detection** 25
  - **3.1.1 Grayscale Conversion** 25
  - **3.1.2 Gaussian Blurring** 27
  - **3.1.3 Histogram Equalization** 28
  - **3.1.4 Thresholding** 30
  - **3.1.5 Morphological Operations** 32
- **3.2 Shape Analysis** 34
  - **3.2.1 Contour detection** 35
  - **3.2.2 Roundness** 35
  - **3.2.3 Bounding Rectangle** 36
- **3.3 Calibration** 36

3.3.1 Importance of calibration 36
3.3.2 Calibration Process 37
3.3.3 Pixels per centimetre 38
3.4 Area, Volume and Mass Calculations.. 38
3.4.1 Area calculations 39
3.4.2 Volume calculation 39
3.4.3 Mass calculation 40
CHAPTER 4 41
SIEVE ANALYSIS 41
4.1 Introduction to Sieve Analysis.. 41
4.1.1 History 41
4.2 Principles of Sieve Analysis 41
4.2.1 Analysis of sieve analysis result 42
4.3 Interpretation of Results 43
4.3.1 Determining D-values 44
4.3.2 Coefficient of Uniformity and Curvature for Soil Classification. 44
CHAPTER 5 47
TECHNOLOGIES USED 47
5.1 Python Libraries.. 47
5.1.1 Open CV 47
5.1.2 Numpy.. 47
5.1.3 Matplotlib 48
5.1.4 Pandas.. 48
5.2 Graphical User Interface (GUI) 49
5.2.3 Tkinter 49
5.3 File Handling 49
5.3.1 Reading images 49
5.3.2 Saving Processed Image.. 49
5.3.3 Data Files.. 50
5.4 Data Export 50
5.4.1 CSV Files.. 50
5.4.2 Graphs and Plots.. 50
CHAPTER 6 51
FROM CODE TO APPLICATION: DETAILED IMPLEMENTATION..51
6.1 Introduction.. 51
6.2 Code Explanation for Image Analysis 51
6.2.1 Importing libraries.. 51
6.2.2 Steps involved.. 51

6.3 Code Explanation for Sieve Analysis Functionality ... 53
6.3.1 File Selection and CSV file import ... 54
6.3.2 Sieve Size Categorization ... 54
6.3.3 Mass calculation and Aggregation ... 54
6.3.4 D-Values Calculation ... 54
6.3.5 Coefficient of Uniformity and Coefficient of Curvature ... 55
6.3.6 Data Visualization ... 55
6.4 Software Functionality Demonstration ... 57
6.4.1 Image Contour Analysis ... 57
6.4.2 Extract Data from Image ... 60
6.4.3 Gradation Curve ... 61
CHAPTER 7 ... 63
SUMMARY CONCLUSION AND FUTURE WORK ... 63
7.1 Summary ... 63
7.2 Conclusion ... 63
7.3 Scope for future Study ... 63
References ... 65

LIST OF FIGURES

Figure 1 (a) Open-pit Mining (b) Strip Mining (c) Mountaintop Removal Mining (d) Dredging................. 13
Figure 2 (a) Rotary Drilling (b) Percussive Drilling (c) Diamond Drilling.. 14
Figure 3 Blasting 15
Figure 4 Loading and Hauling 15
Figure 5 Crushing.. 16
Figure 6 Grinding... 16
Figure 7 Screening and Classification 17
Figure 8 Flowchart for Image analysis of Rock fragment 19
Figure 9 Image before Grayscale 26
Figure 10 Image after Grayscale 26
Figure 11 Gaussian Curve and its effect on blurring image 27
Figure 12 Image after Gaussian Blurring 28
Figure 13 Effect on Histogram during histogram equalization.. 29
Figure 14 Image before Histogram Equalization 29
Figure 15 Image after Histogram Equalization 30
Figure 16 Effect on Histogram during Thresholding 31
Figure 17 Image before Thresholding.. 32
Figure 19 Image after Thresholding... 32
Figure 20 Morphological operations... 34
Figure 21 Image after Morphological Operations 34
Figure 22 Contoured Image 35
Figure 23 Roundness and Area of rocks detected by contouring 35
Figure 24 Bounding Rectangle helps determine height or width of the rock.................................... 36
Figure 25 Ruler as a reference with a known dimension of 15 cm.. 37
Figure 26 Accurate detection of height of rock with respect to the ruler 38
Figure 27 on applying contouring, area of rock is determined 39
Figure 28 Manual sieve Analysis Technique 41
Figure 29 Automate Sieve Analysis Technique 42
Figure 30 OpenCV 47
Figure 31 NumPY... 48
Figure 32 Matplotlib.. 48
Figure 33 Pandas... 48
Figure 34 Graphical User Interface of the Software 49
Figure 35 Important Libraries... 51
Figure 36 Code snip to detect and measure contour 52
Figure 37 Code snip of Grayscale conversion.. 52
Figure 38 Code snip for Gaussian Blurring.. 52
Figure 39 Code snip for Histogram Equalization 52
Figure 40 Code snip for Thresholding.. 52
Figure 41 Code snip for Morphological Transformation 53
Figure 42 Code snip for contour detection.. 53
Figure 43 Output after Contour detection 53
Figure 44 File selection and CSV file import.. 54
Figure 45 Sieve size categorization 54
Figure 46 Mass calculation and Aggregation.. 54
Figure 47 D-values calculation.. 55
Figure 48 Coefficient of uniformity and coefficient of curvature 55
Figure 49 Data visualization... 56
Figure 50 Gradation Curv... 56
Figure 51 Homepage of the Software 57

Figure 52 Select the image 57
Figure 53 Cropping of the selected image 58
Figure 54 Contouring on the rock fragments 58
Figure 55 User satisfied or not 58
Figure 56 If no, image saved in desktop 59
Figure 57 Manual Contouring 59
Figure 58 Red lines shows manual contouring 59
Figure 59 Selecting recently saved Contoured image 60
Figure 60 User enters density value of the rock 60
Figure 61 Generating CSV file 61
Figure 62 Selecting CSV file 61
Figure 63 Final output, i.e. Gradation curve of the rock fragment analysis 62

CHAPTER 1

INTRODUCTION

1.1 Introduction to Mining Operations

Mining plays an integral role in the global economy, all necessary raw materials required for various industries are provided by this field of engineering i.e. Mining. Mining involves extraction of valuable minerals from the Earth. These materials include coal, metal, oil shale, gemstones, limestone, rock salt, potash, clay etc. For successful extraction and processing of mineral, mining involves several stages for that purpose.

Surface Mining is a type of Mining operations. Surface mining is a method of extraction of ore and minerals located near to Earth's surface. Surface Mining encompasses several technique.

- **Open-pit Mining**: Involves digging a pit or hole in the ground to access desired minerals.
- **Strip Mining**: involves mining coal and lignite. It also involves removing long strips of overburden to expose underlying coal seam.
- **Mountaintop Removal Mining**: This method is used in areas with rugged terrain. It involves removing summit of a mountain to extract coal seams.
- **Dredging**: This method involves technique to mine material from underwater sediments, especially in rivers, lakes or seabed.

Figure 1 (a) Open-pit Mining (b) Strip Mining (c) Mountaintop Removal Mining (d) Dredging

1.2 Unit operations in Mining Engineering

It is a systematic process that involves extracting and processing minerals. Each unit operations collectively contributes in overall mining process.

1. Drilling

The purpose of drilling is to create hole in the rock to place the explosives for blasting to extract samples for analysis. Equipment's used for Drilling are Drill rigs, augers and diamond drills. Different types of drilling methods are:

- **Rotary Drilling**: large diameter holes are created, used to extract oil and gas.
- **Percussive Drilling**: involves repeatedly striking the rock with a bit.
- **Diamond Drilling**: it uses diamond-tipped bits for precise and efficient core sampling.

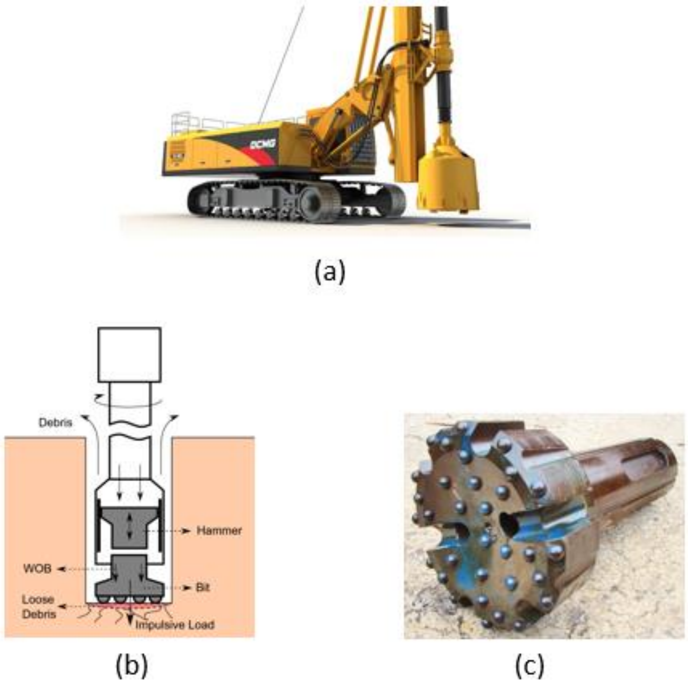


Figure 2 (a) Rotary Drilling (b) Percussive Drilling (c) Diamond Drilling

2. Blasting

For easier handling and transportation blasting is used to break up rocks. The process of Blasting involves drilling holes, Loading Explosives and Detonation. Types of explosives used for Blasting are ANFO (Ammonium Nitrate Fuel Oil), Dynamite and emulsion explosives.

Figure 3 Blasting

3. Loading and Hauling

The purpose of Landing and Hauling is to transport broken rocks or ores from mining site to processing site or waste dump. Equipment's used for this process is Loaders, and Haul Trucks.

Figure 4 Loading and Hauling

4. Crushing

The purpose of this method is to reduce the size of the large rocks and ore to facilitate for further processing. Different types of crushing methods are as follow:

- **Primary Crushing**: Initial stage using jaw crushers.
- **Secondary Crushing**: Further size reduction using cone crusher or impact crusher.

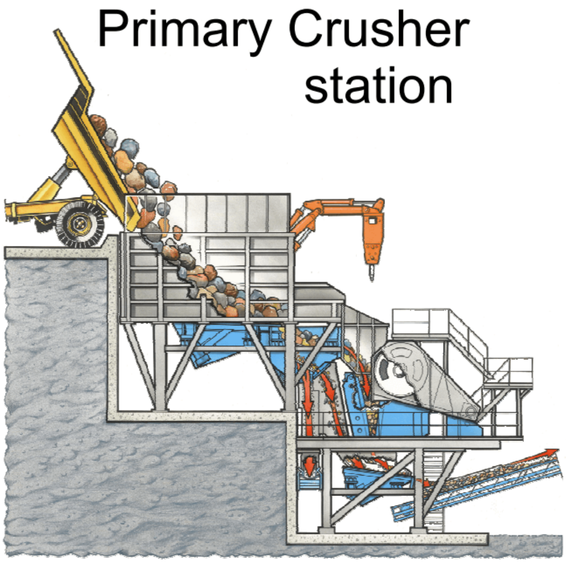


Figure 5 Crushing

5. Grinding

The purpose of Grinding is to further reduce ore size to extract valuable minerals for subsequent concentration process. Different types of grinding methods are:

- **Ball Mills**: Grinding balls are used to crush ore.
- **Rod Mills**: Long rods are used for grinding.
- **SAG Mills**: Combine crushing and grinding.

Figure 6 Grinding

6. Screening and Classification

To separate material based on size for further processing.

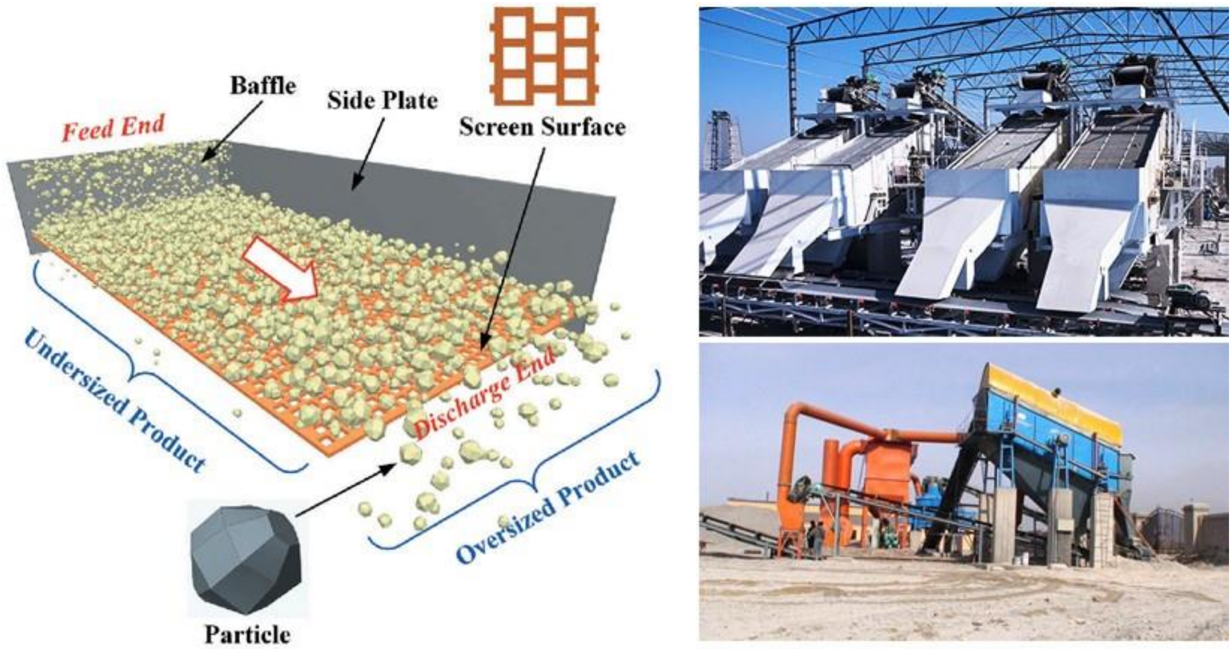


Figure 7 Screening and Classification

1.3 Image analysis in Mining Operations

Image analysis enhances the efficiency and effectiveness of several mining operations, including Blasting, Loading, crushing, grinding and screening and Classification. Here’s a detailed explanation on how image analysis can be applied in each of the process:

1. Blasting

Application of image analysis in blasting:

- **Pre-Blast Analysis:** Identification of natural factures, rock type, and other geographical features can be captured in High-Resolution, this helps in planning the optimal drill pattern and explosive charge distribution.
- **Fragment Analysis:** After blasting, image is analysed and evaluated for size distribution of the blasted fragments.

Benefits of image analysis in Blasting:

- Increases loading efficiency
- Accurate material tracking and reporting and
- Reducing operational cost and time.

2. Loading

Application of image analysis in Loading:

- **Material Identification:** Real-time image analysis can identify different types of material being loaded, this ensures desired material is transported.
- **Volume estimation:** Image can be used to estimate volume of material loaded into haul trucks or conveyors, optimizing loading operations and reducing cycle time.

Benefits of image analysis in Loading:

- Increasing loading efficiency
- Reducing operational cost and time.

3. Crushing

Application of image analysis in crushing:

- **Feed Size Monitoring:** Size distribution of ore can be analysed before feeding into crusher, enabling real-time adjustments to optimize performance.
- **Crusher Wear Monitoring:** Regularly captured image of crusher line can help in monitoring wear pattern, which can help schedule maintenance before major failure.

Benefits of image analysis in Crushing:

- Reduced energy consumption and wear costs
- Improved crusher maintenance

4. Grinding

Application of image analysis in Grinding:

- **Particle Size Distribution**: Image analysis of the grinding mill output can provide detailed information on particle size distribution. This data helps in adjusting grinding parameters to achieve the desired liberation of minerals.
- **Mill Performance Monitoring**: Visual monitoring of the grinding process can detect issues like over-grinding or under-grinding, enabling timely corrective actions.

Benefits of image analysis in Grinding:

- Enhanced grinding efficiency.
- Improved mineral liberation and recovery.
- Reduced energy consumption and operational costs.

5. Image Analysis Application:

Application of image analysis in Grinding:

1. **Screen Efficiency Monitoring**: Image analysis can evaluate the performance of screening operations by assessing the size distribution of materials passing through screens. This helps in optimizing screen aperture sizes and operational settings.

2. **Classification Accuracy**: For classifiers like hydro cyclones and spiral classifiers, image analysis can monitor the size and density of particles, ensuring accurate separation of fines and coarse particles.

Benefits of image analysis in Crushing

- Increased screening and classification efficiency.
- Enhanced control over product quality.
- Reduced loss of valuable minerals in tailings.

1.4 Rock fragment Analysis

Rock fragment analysis is a critical component in industries such as mining, geology, and construction. Accurate measurement and classification of rock fragments can influence operational decisions, safety, and overall project efficiency. Traditionally, this analysis is performed manually, which is time-consuming, labour-intensive, and subject to human error. Recent advancements in image processing and computer vision technologies present an opportunity to automate this process, thereby enhancing accuracy and efficiency.

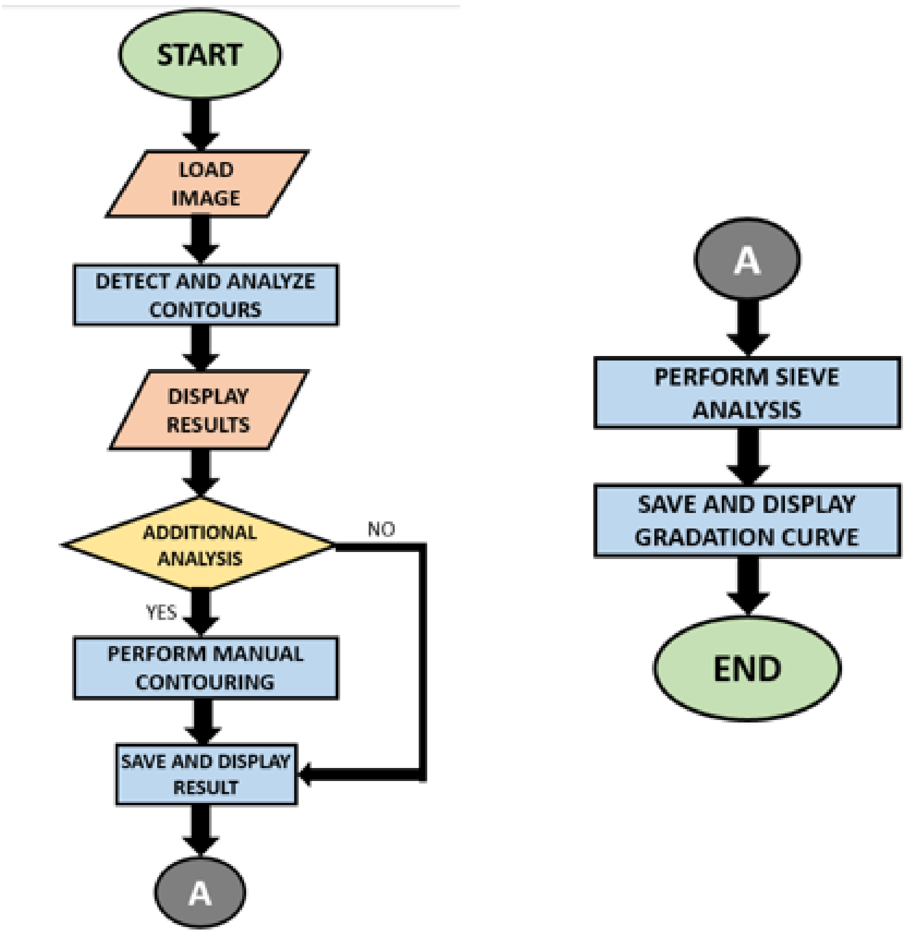


Figure 8 Flowchart for Image analysis of Rock fragment

1.5 Motivation for the Study

The need for an automated rock fragment analysis system is driven by the limitations of manual methods and the potential for improved accuracy, consistency, and efficiency through technological advancements. By leveraging image processing techniques, this project aims to streamline the analysis process, providing rapid and reliable results that can significantly benefit industry operations.

1.6 Problem Statement

Current methods of rock fragment analysis are inadequate due to their dependence on manual measurements, which are prone to errors and inconsistencies. This research aims to develop an automated system to perform accurate and efficient rock fragment analysis using advanced image processing techniques.

1.7 Objectives of the Study

- **Primary Objective**: To develop an automated system for rock fragment analysis using image processing techniques.
- **Secondary Objectives**: To improve measurement accuracy, reduce analysis time, and create a user-friendly interface for practical application.

1.8 Scope of the Study

- **Technical Scope**: This study will focus on the development and application of image processing techniques, including contour detection, homography transformation, and data extraction.
- **Application Scope**: The system will be applicable in the mining, geological, and construction industries for rock fragment measurement and analysis.

1.9 Methodology Overview

The methodology involves acquiring images of rock fragments, pre-processing the images, detecting contours, performing measurements, and analysing the data. The system will be implemented using Python and relevant libraries, with a focus on user-friendly interface design.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the domains of mining engineering, geological engineering, civil engineering, and environmental research, rock fragment analysis is essential. In the past, the size distribution and sorting of rock particles were done by hand, which resulted in labour-intensive tasks, time consumption, and precise inaccuracies. We need a more precise and efficient way to examine the features of rock fragments because the need for them is expanding.

Advancements in machine learning and digital image processing have opened new avenues for rock fragment analysis. These technologies have enabled the potential to automate segmentation and classification of rock specimens, reducing manual intervention and improving the accuracy and efficiency of analysis. Digital image processing allows the extraction of information from images, whereas machine learning models learn from data to make precise predictions about size distribution and sorting level. The involvement of high-resolution imaging and advanced computational techniques has led to the development of sophisticated tools that can process large volumes of image data quickly and accurately. Provide valuable insights that were previously unattainable with traditional methods. For example, convolutional neural networks (CNNs) and support vector machines (SVMs) have shown great improvement in recognizing and classifying rock fragments based on their features, eliminating the need for time-consuming manual segmentation.

The scope of the literature review is to explore the advancements in techniques and their application in fragment analysis. The aim is to provide a concise overview of the current state of knowledge while also highlighting the significant developments in the existing research. This review covers many image processing techniques to improve the performance of analysing rock fragments.

2.2 History

In 1993, a decisive step was taken in the field of rock fragmentation analysis by (John M.Kemeny, 1993) with the addition of digital image processing. This was the beginning of utilizing high-definition video cameras to capture images. To delineate individual rocks accurately, a computer algorithm was designed. This technique allowed the production of size distribution curves from multiple images. Further integration of statistical procedures enhanced the accuracy of image analysis. This method was ground-breaking compared to traditional manual measurements and sieve analysis techniques.

In the year 2012, Image Analysis technique on evaluation of particle size Distribution of gravel (G.H.A. Janaka J. Kumara, 2012), Fast-forwarding to 2012, the field witnessed the introduction of better image analysis techniques. The technique utilized ImageJ, a powerful image analysis software. The study evaluated differences in the gradation curve obtained from conventional methods. In this method, key considerations included minimizing shadow effects and optimizing the conditions under which images were taken. This method only required a camera and a computer, offering a practical and efficient alternative to the traditional method.

In 2018, the notion shifted towards automating the image analysis of rock fragments (Lei Shu, 2018). This method required a new methodology leveraging learning from a deep neural network (CNN) model and a pre-trained large-scale image. The model was then used to train a Support Vector Machine (SVM) classifier. This approach improved the results compared to conventional handcrafted features. This method highlighted the potential of machine learning in the field of rock particle analysis.

2.3 Innovative Advancements in Rock Fragmentation Analysis

There has been a significant advancement in rock fragmentation analysis over the years, from introduction to digital image processing in 1993 to integration of machine learning models in 2018. This project is built on these innovations and introduces several unique features that enhances its capabilities and effectiveness compared to previous work.

1. **Enhanced Image Processing workflow**: This project integrates advanced image processing tools like OpenCV to improve accuracy and contour detection and particle size analysis. On integrating Grayscale conversion, Gaussian Blurring, Histogram Equalization, thresholding, and morphological operations, this ensures precise rock fragment analysis.
2. **User-friendly Interface with GUI:** The software is accessible to users with varying technical background to perform complex analysis with simple interactions.
3. **Automated Data Extraction and Analysis**: This project automates the extraction of relevant data from image and sieve analysis. This includes calculating D-values, coefficients of Uniformity and curvature, and generation of gradation curve.
4. **Data Visualization**: User can generate and view gradation curves. Cumulative passing percentage plots, histograms all these features helps in understanding the particle size distribution.

2.4 Key theories

Theoretical Framework

1. **Image processing:**

   There are several steps applied to enhance the image for contour detection

   1. **Grayscale conversion:** It is one of the conversion method in which the coloured image is converted to grayscale, as grayscale reduces computational complexities by eliminating colour channels and focusing on intensity values.
   2. **Blurring:** it is s procedure which is used to smooth the image and reduce noise, which has the chance it can interfere with contour detection. Gaussian kernel applied in this process spreads pixel values based on normal distribution.
   3. **Histogram Equalization:** Enhances image contrast by redistributing the intensity values. This technology makes the features in the image more distinguishable, which can support for better contour detection.

4. **Thresholding:** this method converts grayscale image into binary image by highlighting the object of interest against the background. Otsu's method is used to determine the optimal threshold value.
5. **Contour Detection:** using algorithms like the border following algorithm contours are detected. These contours are the boundaries of the object and are essential for further analysis.

2. **Shape Analysis:**

   The methodology is used to evaluate the geometric properties of contoured image and understand its physical properties.

   1. **Roundness Factor**: $= \frac{4 \times \pi \times Area}{perimeter^2}$ This formula indicates how approximately the contour is circle.
   2. **Bounding Box and Enclosing Circle**: It provides the smallest rectangle that contain the contour, giving the width and height of the object.

3. **Area, Volume and Mass calculations:**

   To quantify the size of fragments converting pixels to real-world unit is important.

   1. **Area Conversion**: The area of the contour in pixel is converted to square centimetres or square millimetres using a known scale, determined by the pixels per centimetre (or millimetre).
   2. **Volume calculation**: The volume of the fragments is calculated based on their shape. For nearly spherical objects, the volume is computed using the formula for volume of sphere. For more ellipsoidal objects, the formula for the volume of an ellipsoidal is used.
   3. **Mass calculations**: Mass of the fragments is determined using the volume and density value is entered by the user.

4. **Sieve Analysis:**

   This method is used to determine the particle size distribution of a granular material.

   Gradation Curve: This curve is used to plot the distribution of particle size. This is done by categorizing particles into ranges and calculating the cumulative percentage passing through each sieve size.

2.4 Conclusion

On integrating all these theories we develop a tool that provides a solution for analysing image of fragments. This project represents a significant advancement in the field of rock fragment analysis. By integrating cutting-edge image processing techniques, a user-friendly interface, automated data extraction, and robust data visualization capabilities, it offers a powerful and efficient tool for analysing rock fragments. The combination of these features not only improves the accuracy and efficiency of the analysis but also makes it accessible to a broader audience, ensuring its practical application in various domains such as mining, geological, civil, and environmental engineering. This project builds on the foundation of previous work and introduces innovative solutions that address the limitations of traditional methods, paving the way for further advancements in rock fragment analysis.

CHAPTER 3

IMAGE PROCESSING

3.1 Enhancement of images for contour detection

One of the crucial step in Image processing is Image Enhancement, Aim is to enhance the visual appearance of an image for better analysis. In terms of contour detection of fragments, primary goal is to enhance the feature of interest.

3.1.1 Grayscale Conversion

Grayscale Conversion is the first stage of image enhancement, this process is used to simplify the computation complexity, this process simplifies the image data by reducing three colour channel i.e. red, blue, green to a single intensity channel, in this conversion it focuses on structural details of the image. This step makes the process faster and efficient while maintaining critical features (Saravanan, 2010).

Logic:

- **Human Perception of Brightness**: Green light, followed by red light, and then followed by Blue light are the Light colours by wich human eyes are senstive. These considerations are taken while converting the image from color to Grayscale.
- **Weighted Sum Method**: To convert colored image to Grayscaledimage sum of Red, Green and Blue components are calculated for each pixels. Replace RGB values of pixels with single Gray value calculated. .

Mathematics:

$$Gray = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

Here:

R: Red component of the pixel

G : Green component of the pixel

B: Blue component of the pixel.

Coefficients 0.299,0.587 and 0.114 are derived from Luminance component of the YUV color space, which models human Vision more accurately.

In YUV color space

- Y represents Brightness or luminance component.
- U and V represents color components.

Reason behind these coefficents:

- **Green**: Green has the highest coefficent 0.587, human eyes are more senstivie to this color.
- **Red:** Red has the coeffient of 0.299, human eyes are moderately senstive to this color.
- **Blue:** Blue has the least coefficent 0.114, Human eyes are least senstive to this color.

Figure 9 Image before Grayscale

Figure 10 Image after Grayscale

3.1.2 Gaussian Blurring

Gaussian Blurring method is used to reduce noise and smooth out variation in pixel on Grayscale image. This method involves convolving images with Gaussian Blurring detection of the edges of details. Using Gaussian Blurring detection of edges of fragments becomes easier. It is essential for accurate contour detection (Gedraite, 2011).

Logic:

- Gaussian Blurring is based on the principle of Gaussian distribution or normal distribution. This describes how values are distributed around a mean. Which is characterized by Bell-shaped curve in mathematical function.

Mathematics:

- **Gaussian function in One-dimension:**

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2a^2}}$$

- **Gaussian function in Two-dimension:**

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2+y^2}{2a^2}}$$

Here:

1. (x,y) are coordinates of pixel relative to the centre of the kernel
2. σ Is standard deviation, it controls the spread of the Gaussian Curve. Larger the σ value, results in more blurring.

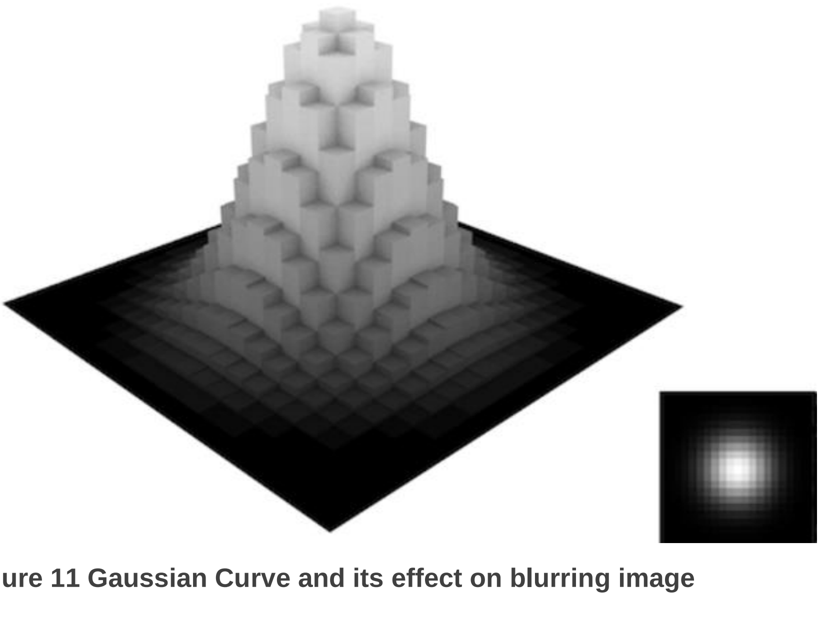


Figure 11 Gaussian Curve and its effect on blurring image

Figure 12 Image after Gaussian Blurring

3.1.3 Histogram Equalization

Histogram Equalization is a technique that is used to enhance the contrast of an image. This method adjusts the intensity distribution of image. Area with poor contrast becomes more distinct. This process highlights edges and boundaries which makes it easier for contour detection (M.Abdullah-Al-Wadud, 2007).

Logic:

- In Histogram Equalization, the logic is to spread out the brightness level evenly across the range from 0 to 255. This enhances the contrast, making dark image darker and bright image brighter.

Mathematics:

- **Compute Histogram:**
  Calculate histogram of the original image
  $$h(i) = number\ of\ pixels\ with\ intensity\ i$$

- **Cumulative Distribution Function:**
  CDF at value 'k' is the sum of the histogram values from lowest intensity up to 'k'.
  $$CDF(k) = \sum_{i=0}^{k} h(i)$$

- **Normalize the CDF:**
  Normalizing CDF to range 0 to 1 by dividing it by total number of pixels N in the image.

$$CDF_{normalized}(k) = \frac{CDF(k)}{N}$$

- **Map Original Intensities :**
  Map original intensity levels to new intensity levels using normalized CDF. The new intensity value for a pixel with original intensity I is given by.
  $$I' = round(CDF_{normalized}(I) \times (L-1))$$
  L is the number of possible intensity levels (256 for an 8 bit image).

- **Creating an Equalized image**
  Replacing pixels intensity in original image with its new intensity value to get equalized image.

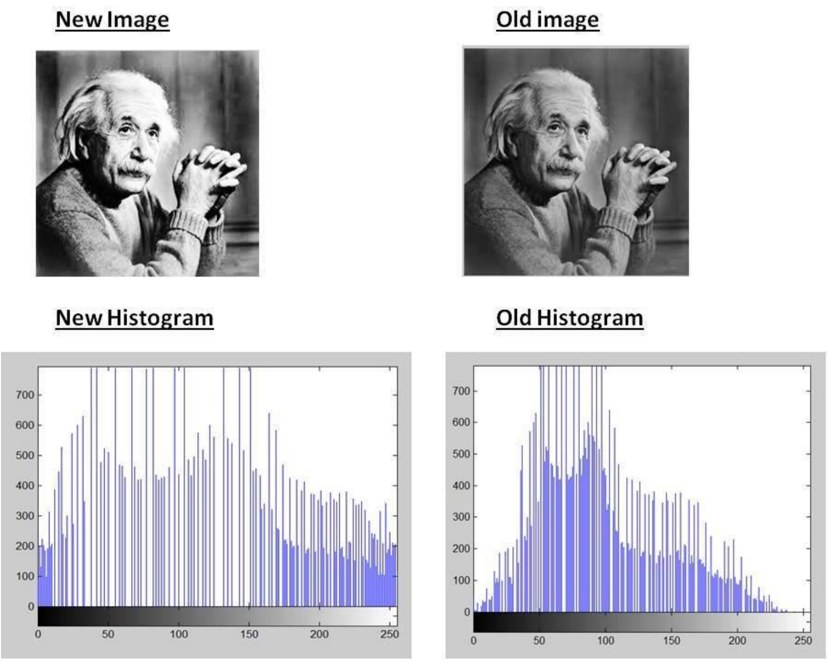


Figure 13 Effect on Histogram during histogram equalization

Figure 14 Image before Histogram Equalization

Figure 15 Image after Histogram Equalization

3.1.4 Thresholding

Thresholding converts grayscale image to binary image, where pixels are either black or white. On applying thresholding, images is divided onto foreground i.e. contour and background regions. This binary representation is necessary for contour detection.

Logic:

- The binary image is obtained by converting all pixels below a certain value to black (0) and all pixel values above this value to white (255).

Mathematics:

Types of Thresholding:

- **Global Thresholding**
  Considering a grayscale image I(x, y), with pixel intensities 0 to 255. We choose a threshold value 'T'. Thresholding can be processed by.
  $$I'(x,y) = \begin{cases} 255 & if\ I(x,y) \geq T \\ 0 & if\ I(x,y) < T \end{cases}$$
  Where I'(x, y) is output binary image.

- **Adaptive Thresholding**
  In adaptive Thresholding (N.B. Rais, 2005) the threshold value is not fixed, it is calculated for each pixel based on the intensity values of the surrounding pixel.
  Using mean value of the neighbourhood.
  $$T(x,y) = \frac{1}{N} \sum_{(i,j) \epsilon neighbor} I(i,j)$$
  N is the number of pixels in the neighbourhood
  The threshold image is then

$$I'(x,y) = \begin{cases} 255 & if\ I(x,y) \geq T(x,y) \\ 0 & if\ I(x,y) < T(x,y) \end{cases}$$

- **Otsu's method**
  Otsu's method (Ta Yang Goh, 2017) automatically determines the threshold value by maximizing the between-class variance
  1. Calculate the histogram of the image
  2. Compute the probabilities of each intensity level.
  $$P(i) = \frac{number\ of\ pixels\ with\ intensity\ i}{total\ number\ of\ pixels}$$
  3. Calculate the cumulative sum and cumulative mean for the intensity level
  $$S_k = \sum_{i=0}^{k} P(i)$$
  $$M_k = \sum_{i=0}^{k} i \cdot P(i)$$
  Here:
  S_k Is cumulative sum
  M_k Is cumulative mean
  4. Compute the between-class variance for each possible threshold 'k'.
  $$\sigma_b^2(k) = \frac{(M_T \cdot S_K - M_K)^2}{S_K \cdot (1 - S_K)}$$
  Here:
  M_T Is the total mean of the image?
  5. The threshold value T is chosen to maximizeσ_b^2.

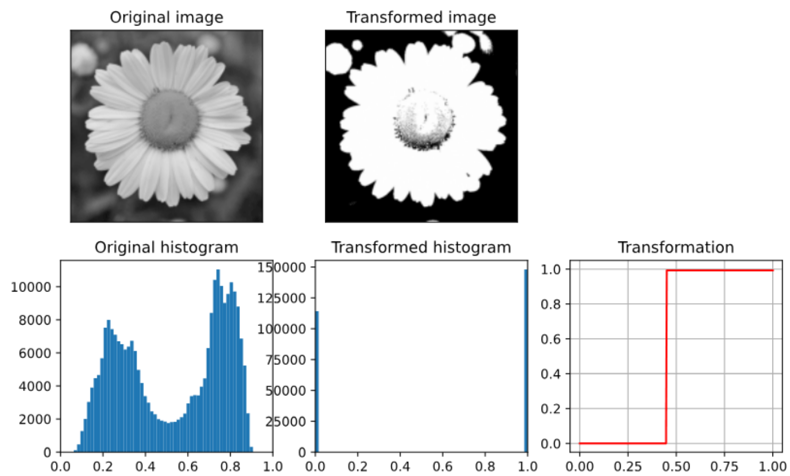


Figure 16 Effect on Histogram during Thresholding

Figure 17 Image before Thresholding

Figure 18 Image after Thresholding

3.1.5 Morphological Operations

In Morphological operations (Mary L. Comer, 1999) process like dilation and erosion are applied to binary image to refine the contour. Dilation tends to expand the boundaries of foreground, by filling small holes and gaps within the contour. Erosion tends to shrink the boundaries of foreground, removing small noises. By smoothing the edges and eliminating noises, this operation enhances the quality of detected contours.

Logic:

Structuring elements is a matrix that determines how morphological operation is applied to each pixel of input image. It consists a pattern of 1's and 0's.

- **Erosion**

Erosion removes pixel around the object boundaries. Makes object in binary images smaller by eroding away the boundaries of white pixels. Useful in removing noise and disconnecting object.

- **Dilation**
  Dilation adds pixels to the boundaries of object in an image. Makes object in binary image larger by dilating the boundaries of white pixels. Useful in joining broken parts, filling holes and expanding objects.

- **Opening**
  It is a combination of erosion followed by dilation. Used to remove small objects or noise from an image, while keeping overall shape and size of larger objects intact.

- **Closing**
  It is a combination of dilation followed by erosion. Used to close small holes and gaps within objects and to connect adjacent objects without changing their size.

- **Gradient**
  Gradient is defined as the difference between dilated and eroded image

Mathematics:

- **Erosion**
  Given binary image I and structuring elements S
  $$I \ominus S = \{z \mid S_z \subseteq I\}.$$
  Here: S_z is the structuring element translated to position z.

- **Dilation** ⍰
  Given binary image I and structuring elements S
  $$I \oplus S = \{z \mid (S_z \cap I) \quad = \emptyset\}$$
  Here: S_z is the structuring element translated to position z.

- **Opening**
  Given binary image I and structuring elements S
  $$I \circ S = (I \ominus S) \oplus S$$
- **Closing**
  Given binary image I and structuring elements S
  $$I \cdot S = (I \oplus S) \ominus S$$
- **Gradient**
  Given binary image I and structuring elements S
  $$G = (I \oplus S) - (I \ominus S)$$

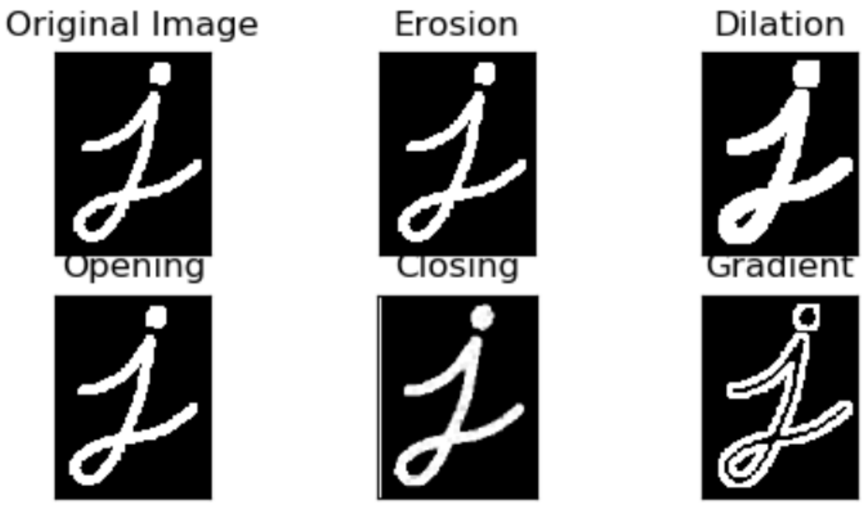


Figure 19 Morphological operations

Figure 20 Image after Morphological Operations

3.2 Shape Analysis

Shape Analysis plays an important aspect in the field of image processing, especially in the field of object recognition, pattern detection, and computer vision. This method involves examination of geometric properties. In this section we will delve into various techniques required for shape analysis.

3.2.1 Contour detection

This technique identifies the boundaries of object within an image. Contour is a curve that joins all continuous points along the boundary that have same colour and intensity. This feature allows isolation and extraction of feature of object. Process starts from Grayscale conversion, Gaussian Blurring, Histogram Equalization and Morphological operation. Contour allows to understand the properties and structure of fragments in the image.

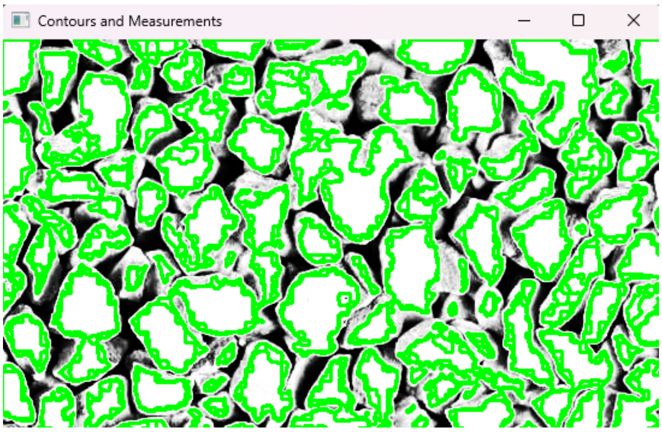


Figure 21 Contoured Image

3.2.2 Roundness

Roundness measures if the object approaches to be a perfect circle. Roundness of object can be calculated using formula

$$Roundness\ Factor = \frac{4 \times \pi \times Area}{perimeter^2}$$

Irregular shaped fragments have roundness factor less than 1, Perfect circle have roundness of 1. This is useful in differentiating between circular and non-circular objects in image.

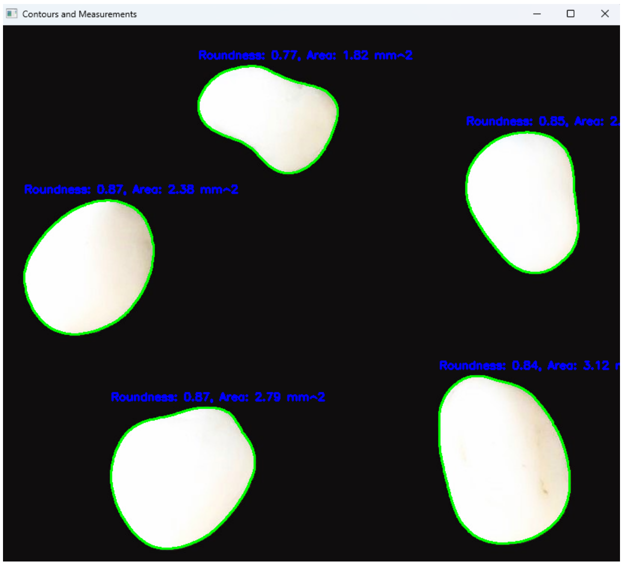


Figure 22 Roundness and Area of rocks detected by contouring

3.2.3 Bounding Rectangle

Bounding rectangle is the smallest rectangle that bounds the largest contour. It is powerful tool in shape analysis that provides the orientation of objects. There are two types of bounding rectangles: Axis- aligns and rotated, in axis-aligned bounding rectangles has sides parallel to image, whereas, in rotated bounding rectangle have sides at any angle.

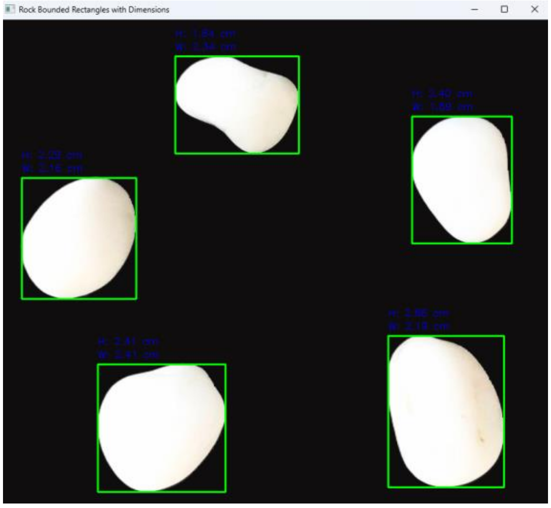


Figure 23 Bounding Rectangle helps determine height or width of the rock

3.3 Calibration

In image processing calibration plays an important role because it involves in establishing relationship between images pixel measurement and real world unit. This process makes sure that measurement are accurate and reliable, this enables precision based analysis and interpretation of object with image.

3.3.1 Importance of calibration

Calibration enables conversion of pixel based measurements to real world unit. Without calibration measurement taken will be meaningless. For example in any field be it medical or engineering, measurement plays a vital role in diagnosing and planning. By calibration we can achieve accurate measurements, enhancing the image analysis result.

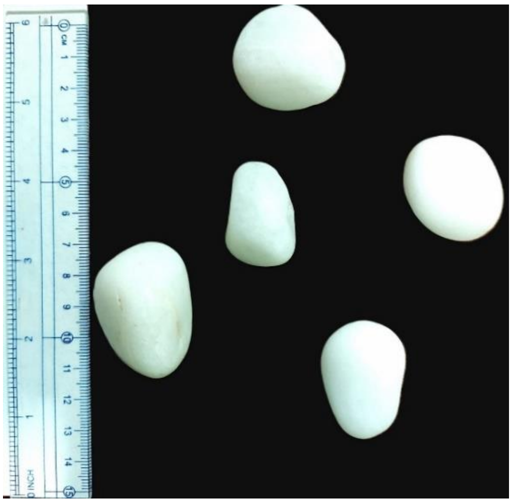


Figure 24 Ruler as a reference with a known dimension of 15 cm

3.3.2 Calibration Process

Calibration involves following steps:

1. **Capturing Calibration Image:** First process starts with capturing an image of reference object. Such as ruler or calibration grid with known dimension. These reference serves to establish a relationship between pixel measurement and real world measurement.
2. **Identifying Reference Point:** In the captures image, reference points are identified. Example edges or corners on calibration grid can serve as reference point.
3. **Calculating pixel to unit conversion factor:** The distance between reference points in pixels are measured and compared to real-world distances. This gives conversion factor, it is often expresses in terms of pixel per centimetre or pixel per millimetre.

By following these steps, the image measurements taken are accurate.

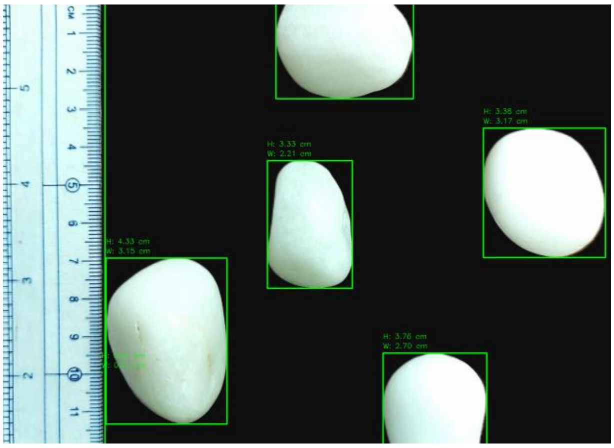


Figure 25 Accurate detection of height of rock with respect to the ruler

3.3.3 Pixels per centimetre

Pixels per centimetre is an important parameter in calibration, it represents number of pixels corresponding to one centimetre in real world.

$$Pixels/cm = \frac{Number\ of\ Pixels}{Real - world\ distance\ in\ cm}$$

If reference object in an image is 800 pixels in length and actual length of object is 10cm, pixel/cm value is

Pixels/cm $= \frac{800\ Pixels}{10\ m} = \frac{80 pixel}{cm}$

It means every centimetre in real world are of 80 pixels.

3.4 Area, Volume and Mass Calculations

Area, Volume and Mass calculation deals with quantifying geometrical properties of objects. This process is vital where understanding physical dimension and properties of object is essential. In the field of geological analysis it is important to determine the size, shape and mass of rock in resource estimation and exploration. Advancement in Artificial Intelligence and Machine Learning have improved the technique that enabled automated high precise calculations.

3.4.1 Area calculations

Enclosed region of an object is determined by area calculation, this is performed by contour analysis where boundary of object is traced. The area is determined from the properties of the contour

$$Area = \int_{contour} 1dA$$

Area (Pixel) = Number of pixels inside the contour.

In order to convert area from pixel square to physical unit, apply the calibration factor.

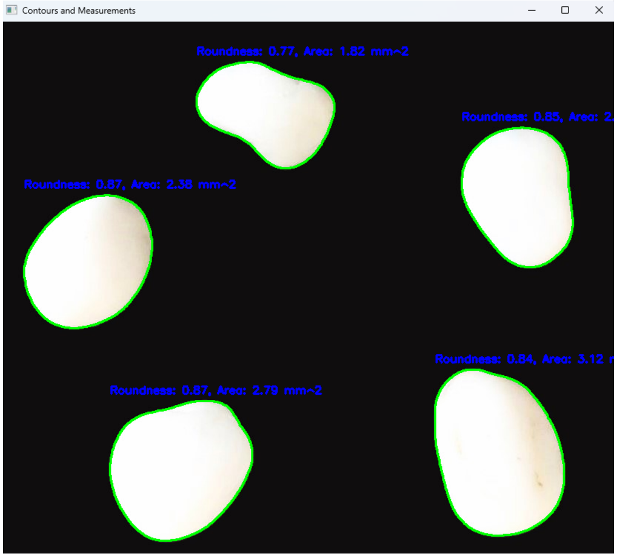


Figure 26 on applying contouring, area of rock is determined

3.4.2 Volume calculation

Estimation of 3D space occupied by an object is determined by volume calculation. Volume calculation for sphere or regular shape is calculated by.

$$Volume = \frac{4}{3}\mu r^3$$

Here, r is the radius of spherical rock.

When object's shape is irregular, volume estimation can be complex. Measurement like average diameter and height obtained from image can be used to determine irregularity volume. Concept of minimum enclosing circle or ellipsoidal can provide rough estimation of object's volume. Other method involves dividing objects into smaller segments, calculating each segment and summing volume to get overall estimate.

If rock is roughly ellipsoidal, volume can be calculated using formula.

$$Volume = \frac{4}{3}\mu abc$$

a, b and c are semi-major axes of ellipsoidal.

3.4.3 Mass calculation

Determination of amount of material within an object is done by mass calculation in kilograms. Mass calculation is closely related to volume calculation, where density of material is provided by user.

$$Mass = Volume \times Density$$

CHAPTER 4

SIEVE ANALYSIS

4.1 Introduction to Sieve Analysis

Determination of particle size distribution of granular material is sieve analysis, also known as gradation testing. This process involves passing the fragments through series of sieve with smaller mesh size and measuring the weight of retained material on each sieve. The main purpose of sieve analysis is to quantify and classify particle size which is important for understanding physical properties and behaviour of fragments in many applications

Figure 27 Manual sieve Analysis Technique

4.1.1 History

The concept of sieve analysis took place during ancient times when traditional sieving technique were purposed in the field of agriculture to separate grains of distinct size. During industrial revolution more standardized and precise sieving method was introduced when need for quality in manufacturing became evident. Over the years, the technique of sieve analysis has evolved to a standardized procedure, making it a fundamental technique in the field of mining and geological engineering.

4.2 Principles of Sieve Analysis

Sieve analysis (FG Carpenter, 1950) operates on principle where segregation of rock fragments based on their size using a series of sieve with small openings. On shaking the sample of granular material through a stack of sieve, each with different mesh size to segregate the particle. The retained amount of weight on each sieve helps in determination of particle size distribution.

4.2.1 Analysis of sieve analysis result

Once the weight of material retained on each sieve is recorded, data is analysed in order to determine particle size distribution. This procedure involves several steps

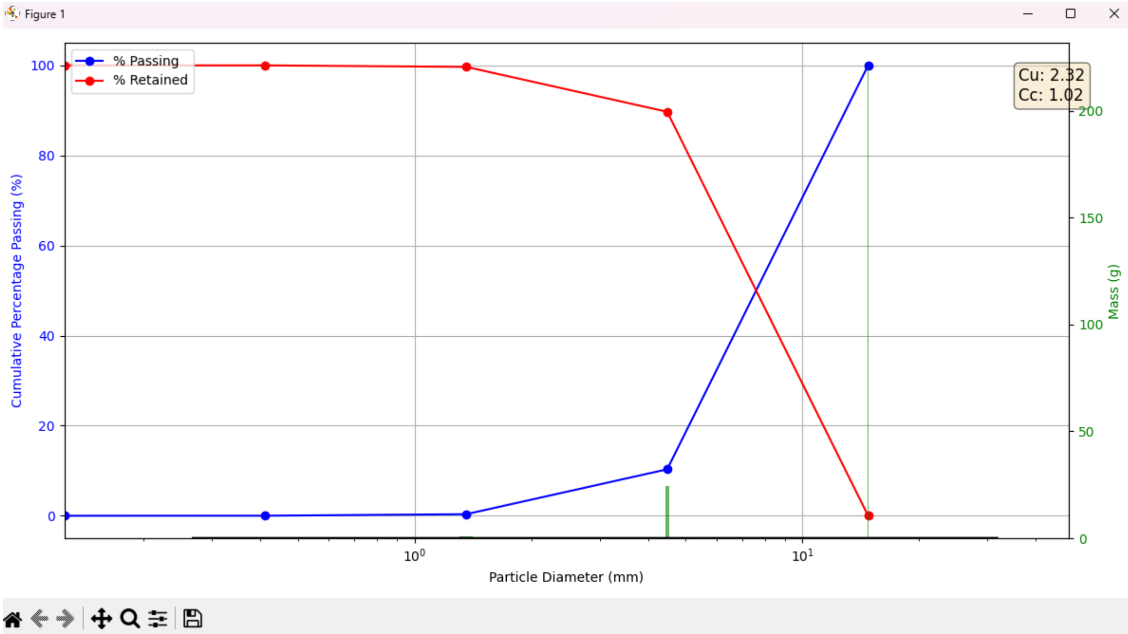


Figure 28 Automate Sieve Analysis Technique

- **Calculation of percent retained**

  First step involves calculations of percentage of total weight sample that retained on each sieve.

$$\text{Percent Retained} = \left(\frac{weight\ Retained}{Total\ sample\ weight}\right) \times 100\%$$

  Example: if 600 gram of sample has 200 gram retained on 2mm, the percent retained is.

$$\text{Percent Retained} = \left(\frac{200}{600}\right) \times 100\% = 33.3\%$$

- **Calculation of cumulative Percent Passing**

  Calculation of cumulative percent passing is essential to understand the distribution of smaller particles in the sample. This is calculated by subtracting the cumulative percent retained from 100%.

  Cumulative percent passing = 100−Ccumulative percent retained

Cumulative percent is obtained by summing the percent retained starting from the coarsest sieve to the one being considered. If the percent retained of first sieve is 15%, and second sieve is 25%, then cumulative percent retained on second sieve is:

$$15\% + 25\% = 40\%$$

Therefore, cumulative percent passing for the second sieve is:

$$100\% - 40\% = 60\%$$

- **Plotting of Gradation curve**

  Gradation curve is required to visualize particle size distribution, in the plotting of gradation curve, the X- axis is represents sieve size, it is usually represented in logarithmic scale, and Y-axis represents cumulative percent passing. The sieve size decreases from left to right and the cumulative percent passing increases from bottom to top.

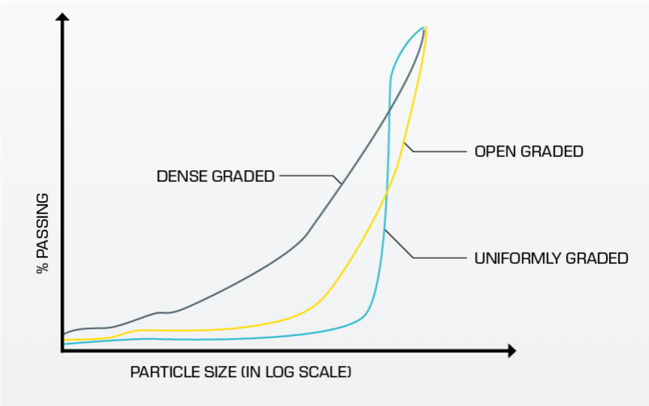
  On connecting the data points, we discover a curve that how particles in the sample are distributed in distinct size ranges. The shape and slop of gradation curve provides materials characteristics.

4.3 Interpretation of Results

- **Plotting the curve**: In gradation curve X-axis represents sieve size, while cumulative percent passing is represented on Y-axis. On the connection of data point, a smooth curve is obtained.

- Interpretation of the shapes
  - **Steep slope:** uniformly graded soil with narrow range of particle size.
  - **Gradual slop**: well graded soil with broad range of particle size.
  - **Flat section**: poor gradation.

- Characteristics observed from gradation curve
  - **Soil Behaviour Prediction:** Under soils permeability and it strength, the shape of curve helps in predicting soils behaviour.
  - **Particle Size Distribution:** smoother the curve, well-graded soil. If curve is steeper, more abrupt curve then it is uniformly graded soil.



4.3.1 Determining D-values

D-values are specific particle diameters on the gradation curve at which a certain percentage of sample's total weight passes through the sieve.

- **D10:** 10% of soil sample passes at this diameter, it is the effective size and also indicated the soil's permeability.
- **D30:** 30% of soil sample passes, used in calculation of coefficient of curvature.
- **D60:** 60% of soil sample passes. Represents soil's average particle size and used for calculation of coefficient of uniformity.

4.3.2 Coefficient of Uniformity and Curvature for Soil Classification.

Key parameters that are used to distinguish soils based on their particle size distribution is done by coefficient of Uniformity (Cu) and Coefficient of curvature (Cc) (Kennedy C. Onyelowe, 2021).

- **Coefficient of Uniformity (Cu)**

  Coefficient of uniformity measures the soil gradation, this indicates how varied the particles sizes are in the sample.

  Formula: $Cu= \frac{D60}{D10}$

  Here,

  D10: 10% of soil sample passes at this diameter.

  D60: 60% of soil sample passes at this diameter.

  Interpretation:

  1. **Cu>4:** This represents well-graded soil.
  2. **Cu$\leq$ 4**: this represents uniformly graded soil.

- **Coefficient of Curvature (Cc)**

  Coefficient of curvatures enables the idea about the shape of particle size distribution curve.

  Formula: $Cc=\frac{(D30)^2}{(D10\times d60)}$

  Here,

  D30: 30% of soil sample passes at this diameter.

Interpretation:

1. **1≤ $\boldsymbol{Cc} \leq \mathbf{3}$**: Indicates Well-graded soil.
2. **Cc beyond the range:** poorly graded soil or presence of gap in the particle.

CHAPTER 5

TECHNOLOGIES USED

5.1 Python Libraries

For image analysis and data analysis extensive range of python libraries ar4e used. In this project we used python libraries for various task like data manipulation, image processing and visualization.

5.1.1 Open CV

Open CV (open source computer vision) it is an open-source computer vision and Machine Learning library that includes several of computer vision algorithms. The veracity of OpenCV makes it an indispensable tool in the field of image processing (Ivan Culjak, 2012) .

It is mainly recognized for real-time application. The library performs various functions, it facilitates the conversion of image to grayscale. Noise reduction technique like Gaussian Blurring to make edge detection more effective. For identification and extraction of shape within an image contour detection is a critical function provided by OpenCV.

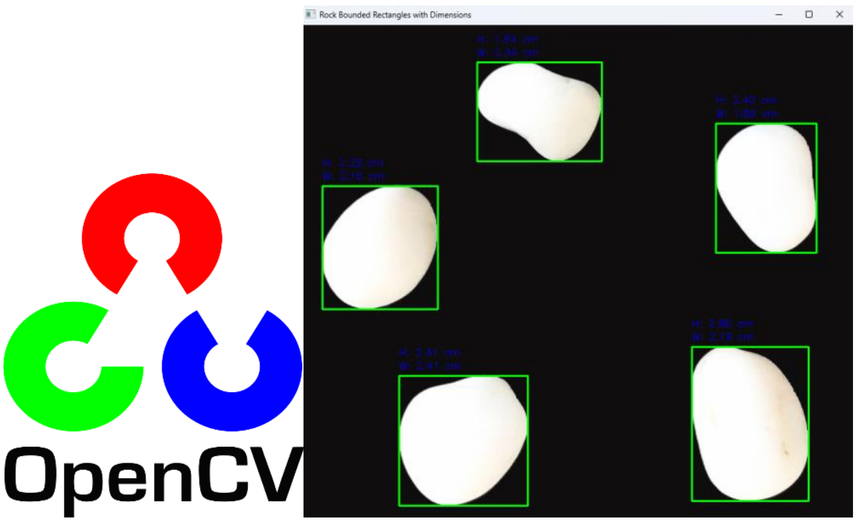


Figure 29 OpenCV

5.1.2 Numpy

For scientific computing with python, Numpy is a fundamental package, Numpy supports collection of mathematical function, large multi-dimensional arrays and matrices. Numpy is used for numerical operations and array manipulation (Stefan van der Walt, 2011). Numpy has the ability to handle large datasets this makes Numpy crucial for processing pixel data of image and performing mathematical operations such as calibration and contour analysis.

Figure 30 NumPY

5.1.3 Matplotlib

Matplotlib is a numerical extension of Numpy, it is a plotting library of python. In this Matplotlib is used for data visualization example, gradation curve in sieve analysis. Complex data can be easily visualized using Matplotlib (Moruzzi, 2020).

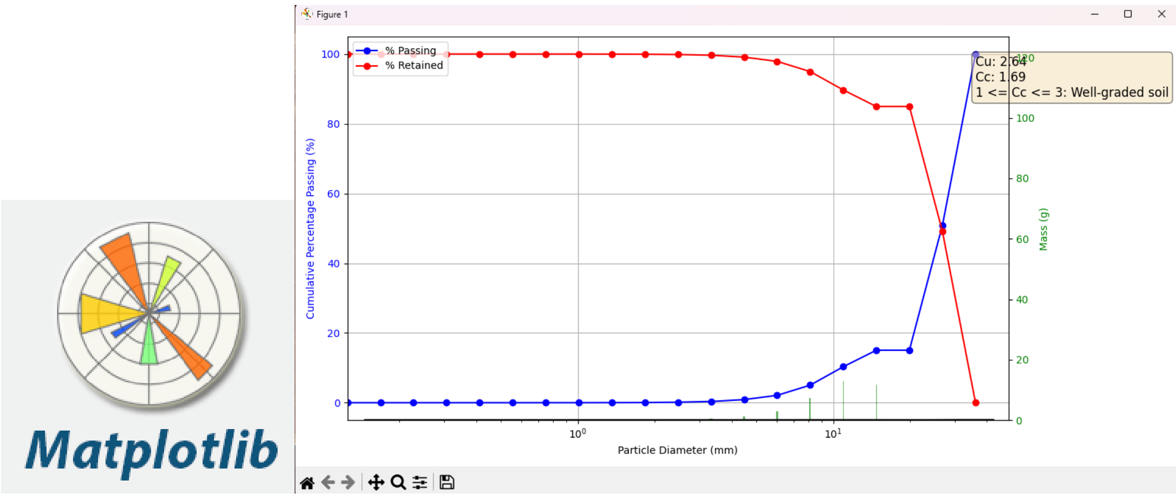


Figure 31 Matplotlib

5.1.4 Pandas

Pandas (McKinney, 2011) is used for handling and manipulating data, it provides data structures like Data Frames and allows efficient Data Manipulation, cleaning and analysis. Pandas have the ability to read and write to various files format, such as CSV, Excel and JSON, makes Pandas a versatile tool for managing data.

Figure 32 Pandas

5.2 Graphical User Interface (GUI)

Graphical User Interface allows the user to interact with the software in a user-friendly manner. GUI framework is used to provide interface for loading images, adjusting parameters and visualizing the results.

5.2.3 Tkinter

The GUI library used in python is Tkinter (Moore, 2018). GUI application are easily created using Tkinter. Customized use interface can be easily created by Tkinter because of its simplicity and flexibility. Without the need of writing of a code user can easily load, adjust the parameter and view the result. For interactive control over the image analysis process Tkinter widgets, such as buttons, sliders, and canvases. This enhances user's experience.

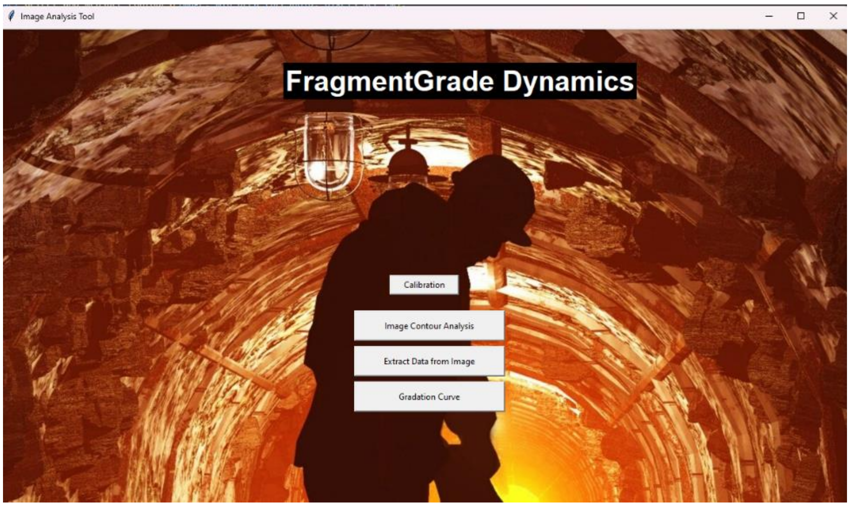


Figure 33 Graphical User Interface of the Software

5.3 File Handling

To manage input and output of data efficiently, file handling is crucial.

5.3.1 Reading images

The 'cv2.imread ()' function is used to load images into application for processing. Various image formats are supported by this function and also ensuring compatibility with different image sources.

5.3.2 Saving Processed Image

The 'cv2.imwrite ()' function saves the processed image, this allows the user to store the result and can analyse the result for future.

5.3.3 Data Files

Data related to sieve analysis and other measurements are written to CSV file using Pandas library. This enables easy data manipulation.

By implementing this file handling, data flows smoothly in different stages for analysis. This also enables proper data backup and recovery.

5.4 Data Export

Data generated from sieve analysis can be exported in various format. Data export is essential for sharing results and further analysis.

5.4.1 CSV Files

Comma separated values, this format is widely used for data manipulation in spreadsheets like Microsoft Excel. Pandas library is used to export data to CSV files. Pandas has the ability to automate data export process, and reduce the risk of error during data transfer.

5.4.2 Graphs and Plots

Graphs and plots are generated using Matplotlib, These files are saved in image format i.e. PNG, JPG. Saving plots in various formats ensures compatibility with different document types.

Each technology plays a crucial role in different stage of this project. The combination of python libraries, GUI, File handling and Data Export creates an effective environment for advanced image processing and sieve analysis.

CHAPTER 6

FROM CODE TO APPLICATION: DETAILED IMPLEMENTATION

6.1 Introduction

The idea of the chapter is to bridge the gap between theoretical knowledge and practical applications. This chapter not only highlights technical complexities but also makes the reader understand the user interaction. The chapter will cover how image is imported to manual contouring by user to Data Extraction and finally followed by sieve Analysis. The goal is to provide clear understanding of the implementation.

6.2 Code Explanation for Image Analysis

The aim is to provide a clear understanding of step involved in image analysis, from importing important libraries to displaying output. We will cover the logic behind each concepts used in this project and demonstrate how the overall software works.

6.2.1 Importing libraries

The first step involves importing all the necessary libraries required for image analysis i.e. OpenCV, Numpy and Matplotlib.

```
from tkinter import Canvas, simpledialog
import cv2
import numpy as np
import os
import pandas as pd
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
import matplotlib.pyplot as plt
```

Figure 34 Important Libraries

- **OpenCV:** this library enables image manipulation, analysis, and filtering
- **Numpy :** this library is used for numerical operations, handling image data in arrays
- **Matplotlib:** this library helps in data visualization.

6.2.2 Steps involved

'detect_and_measure_contours' is function that enables detection of contours, filter the image, and measure their dimensions.

```python
def detect_and_measure_contours(image, min_area_threshold, pixels_per_cm):
    results = []
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    equalized = cv2.equalizeHist(blurred)
    _, thresh_white = cv2.threshold(equalized, 240, 255, cv2.THRESH_BINARY)
    _, thresh_gray = cv2.threshold(equalized, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    kernel = np.ones((3, 3), np.uint8)
    morphed_white = cv2.morphologyEx(thresh_white, cv2.MORPH_CLOSE, kernel)
    morphed_gray = cv2.morphologyEx(thresh_gray, cv2.MORPH_CLOSE, kernel)
    contours_white, _ = cv2.findContours(morphed_white, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    contours_gray, _ = cv2.findContours(morphed_gray, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

    for contour in contours_white + contours_gray:
        roundness = calculate_roundness(contour)
        area_px = cv2.contourArea(contour)
        area_cm2 = convert_to_cm2(area_px, pixels_per_cm)
        if roundness is not None and area_cm2 is not None and area_px >= min_area_threshold:
            cv2.drawContours(image, [contour], -1, (0, 255, 0), 2)
            x, y, w, h = cv2.boundingRect(contour)
            width_cm = w / pixels_per_cm
            height_cm = h / pixels_per_cm
            results.append(f'Roundness: {roundness:.2f}, Area: {area_cm2:.2f} cm^2, Width: {width_cm:.2f} cm, Height: {height_cm:.2f} cm')
    return image, results
```

Figure 35 Code snip to detect and measure contour

- ## Grayscale conversion

Grayscale simply the process by reducing the complexity to a single channel.

```python
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

Figure 36 Code snip of Grayscale conversion

- ## Gaussian blurring

Gaussian blurring smooth the image, reduce noise.

```python
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
```

Figure 37 Code snip for Gaussian Blurring

- ## Histogram Equalization

Enhances the contrast of grayscale image.

```python
equalized = cv2.equalizeHist(blurred)
```

Figure 38 Code snip for Histogram Equalization

- ## Thresholding

Create binary image where pixels are either black or white.

```python
_, thresh_white = cv2.threshold(equalized, 240, 255, cv2.THRESH_BINARY)
_, thresh_gray = cv2.threshold(equalized, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

Figure 39 Code snip for Thresholding

- **Morphological Transformation**

  Applied to threshold image to close small holes and gaps.

```
kernel = np.ones((3, 3), np.uint8)
morphed_white = cv2.morphologyEx(thresh_white, cv2.MORPH_CLOSE, kernel)
morphed_gray = cv2.morphologyEx(thresh_gray, cv2.MORPH_CLOSE, kernel)
```

Figure 40 Code snip for Morphological Transformation

- **Contour Detection**

  Create boundaries to the object in the image.

```
contours_white, _ = cv2.findContours(morphed_white, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
contours_gray, _ = cv2.findContours(morphed_gray, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
```

Figure 41 Code snip for contour detection

Output:

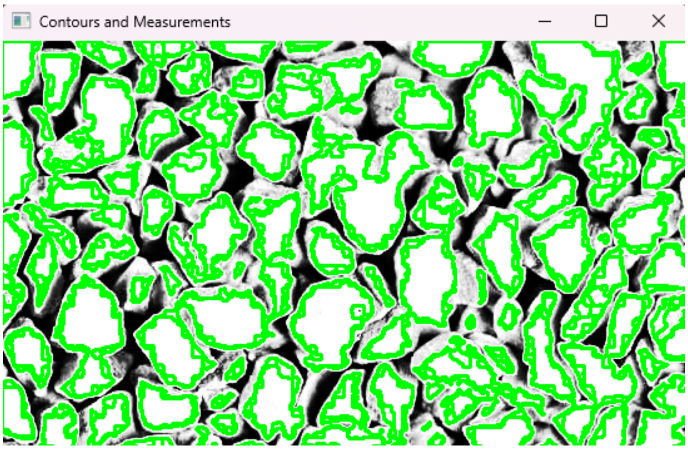


Figure 42 Output after Contour detection

6.3 Code Explanation for Sieve Analysis Functionality

'perform_sieve_analysis' function is used to facilitate the analysis of particle size distribution. In this function the CSV file of particle diameter and masses are imported and visualize the result via Gradation curve.

6.3.1 File Selection and CSV file import

This function asks the user to select The CSV file that contains the required data. Pandas Data Frame reads the selected CSV file.

```
def perform_sieve_analysis():
    root = tk.Tk()
    root.withdraw()
    file_path = filedialog.askopenfilename(title="Select CSV File", filetypes=[("CSV Files", "*.csv")])
    if not file_path:
        print("No file selected. Exiting the function.")
        return

    df = pd.read_csv(file_path)

    # Check if required columns exist
    if 'Average Diameter (mm)' not in df.columns or 'Mass (g)' not in df.columns:
        raise ValueError("The required columns are not present in the CSV file")
```

Figure 43 File selection and CSV file import

6.3.2 Sieve Size Categorization

Range of average diameter is determined, then the range is divided into 20 logarithmic intervals to analyse sieve sizes.

```
# Find the range of average diameter and divide it into a scale of 20
min_diameter = df['Average Diameter (mm)'].min()
max_diameter = df['Average Diameter (mm)'].max()
bin_edges = np.logspace(np.log10(min_diameter), np.log10(max_diameter), num=21)  # 20 intervals

# Categorize average diameters into sieve sizes
df['Sieve Size (mm)'] = pd.cut(df['Average Diameter (mm)'], bins=bin_edges, labels=bin_edges[:-1], include_lowest=True)
```

Figure 44 Sieve size categorization

6.3.3 Mass calculation and Aggregation

The total mass of the sample is calculated. The data is then aggregated by sieve size, summing the masses and calculating cumulative masses and percentages.

```
# Calculate the total mass
total_mass = df['Mass (g)'].sum()

# Aggregate data for sieve summary
sieve_summary = df.groupby('Sieve Size (mm)').agg({'Mass (g)': 'sum'}).sort_index()
sieve_summary['Cumulative Mass (g)'] = sieve_summary['Mass (g)'].cumsum()
sieve_summary['% Passing'] = sieve_summary['Cumulative Mass (g)'] / total_mass * 100
sieve_summary['% Retained'] = 100 - sieve_summary['% Passing']
```

Figure 45 Mass calculation and Aggregation

6.3.4 D-Values Calculation

D-values are specific particle diameters on the gradation curve at which a certain percentage of sample's total weight passes through the sieve.

```python
# Function to find D10, D30, and D60
def find_D_value(percent_passing, target_percent):
    return np.interp(target_percent, percent_passing, bin_edges[:-1])

# Find D10, D30, and D60
D10 = find_D_value(sieve_summary['% Passing'], 10)
D30 = find_D_value(sieve_summary['% Passing'], 30)
D60 = find_D_value(sieve_summary['% Passing'], 60)
```

Figure 46 D-values calculation

6.3.5 Coefficient of Uniformity and Coefficient of Curvature

Coefficient of Uniformity helps in determining if soil is well-graded or uniformly graded

Coefficient of Curvature further helps in classifying well-graded soil.

```python
# Calculate the coefficient of uniformity (Cu)
Cu = D60 / D10

# Calculate the coefficient of curvature (Cc)
Cc = (D30 ** 2) / (D10 * D60)

# Display the coefficient of uniformity and coefficient of curvature with interpretation
print(f"Coefficient of Uniformity (Cu): {Cu:.2f}")
print(f"Coefficient of Curvature (Cc): {Cc:.2f}")

# Interpret the Coefficient of Uniformity (Cu)
if Cu == 1:
    print("Cu = 1 indicates a soil with only one grain size.")
elif 2 <= Cu <= 3:
    print("Cu between 2 and 3 indicates very poorly graded soils, such as beach sands.")
elif Cu >= 15:
    print("Cu of 15 or greater indicates very well graded soils.")
elif 400 <= Cu <= 500:
    print("Cu between 400 and 500 indicates sizes ranging from large boulders to very fine-grained clay particles.")
else:
    print("Cu indicates a soil with moderate grading.")
```

Figure 47 Coefficient of uniformity and coefficient of curvature

6.3.6 Data Visualization

This function creates a gradation curve.

```python
# Plotting the gradation curve and histogram
fig, ax1 = plt.subplots(figsize=(12, 6))

# Plotting Passing Percentage Curve
ax1.plot(bin_edges[:-1], sieve_summary['% Passing'], marker='o', linestyle='-', color='b', label='% Passing')
ax1.set_xlabel('Particle Diameter (mm)')
ax1.set_ylabel('Cumulative Percentage Passing (%)', color='b')
ax1.tick_params(axis='y', labelcolor='b')
ax1.grid(True)
ax1.invert_xaxis()  # Ensure the x-axis shows from maximum to 0
ax1.set_xscale('log')  # Set x-axis to logarithmic scale
ax1.set_xlim(left=bin_edges.min(), right=bin_edges.max())

# Plotting Retained Percentage Curve
ax1.plot(bin_edges[:-1], sieve_summary['% Retained'], marker='o', linestyle='-', color='r', label='% Retained')
ax1.legend(loc='upper left')

# Adding annotations for Cu and Cc
textstr = '\n'.join((
    f'Cu: {Cu:.2f}',
    f'Cc: {Cc:.2f}'))

# These are matplotlib.patch.Patch properties
props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)

# Place the text box in the plot
ax1.text(0.95, 0.95, textstr, transform=ax1.transAxes, fontsize=12,
         verticalalignment='top', bbox=props)

# Plotting histogram
ax2 = ax1.twinx()
ax2.bar(sieve_summary.index.astype(float), sieve_summary['Mass (g)'], width=0.1, alpha=0.6, color='green', label='Mass (g)')
ax2.set_ylabel('Mass (g)', color='g')
ax2.tick_params(axis='y', labelcolor='g')

# Adding a distribution curve to the histogram
mass_distribution = np.repeat(sieve_summary.index.astype(float).values, sieve_summary['Mass (g)'].astype(int))
density, bins, _ = ax2.hist(mass_distribution, bins=bin_edges, density=True, alpha=0)
bin_centers = 0.5 * (bins[:-1] + bins[1:])
ax2.plot(bin_centers, density, color='black', linewidth=2, label='Distribution')

fig.tight_layout()
plt.show()
```

Figure 48 Data visualization

Output:

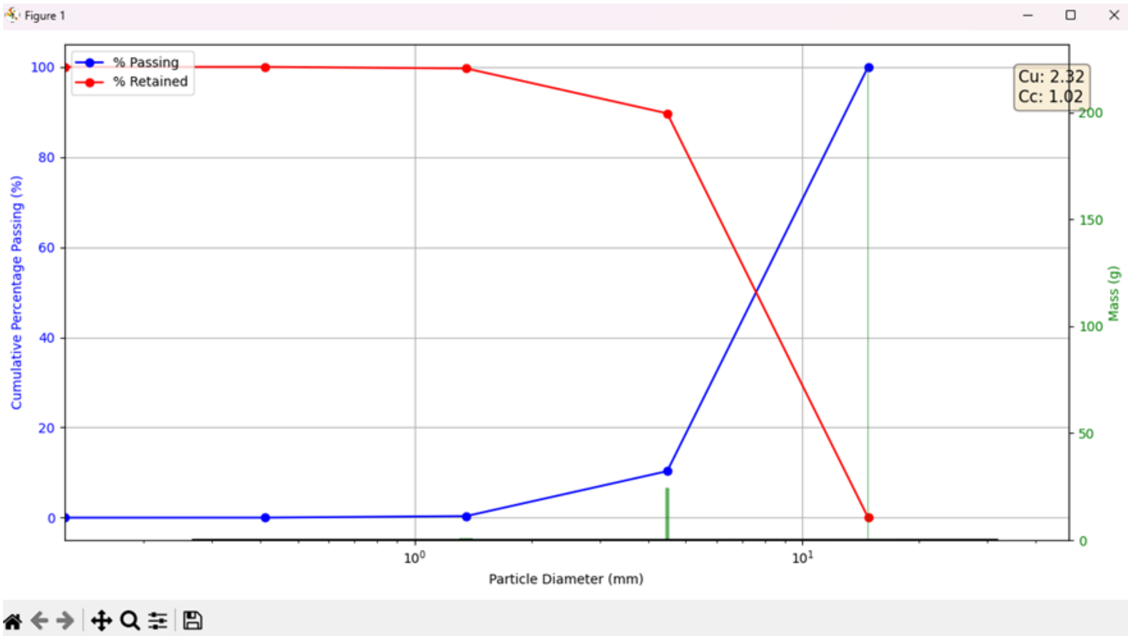


Figure 49 Gradation Curve

6.4 Software Functionality Demonstration

This section is about step-by-step demonstration of the software. This software has three parts firstly, Image contour analysis, secondly, Extraction of data from image then lastly, Gradation curve. These are user-friendly functionalities.

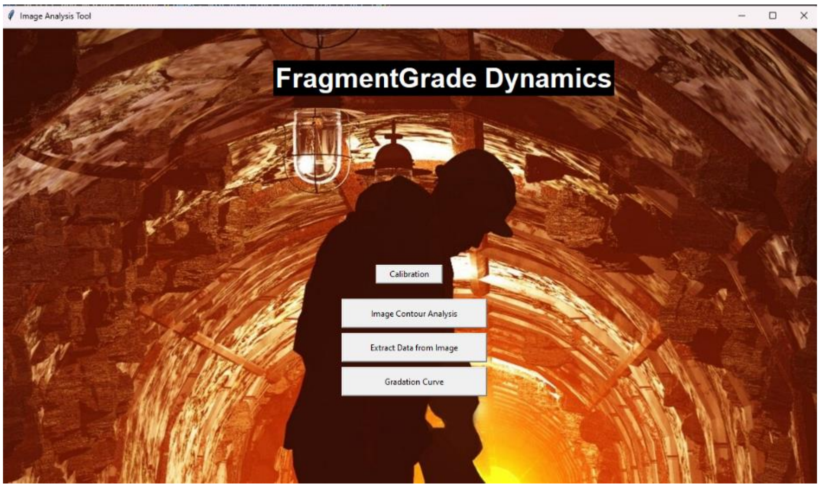


Figure 50 Homepage of the Software

6.4.1 Image Contour Analysis

- **Step 1: Selecting the Image**

  A file dialog box appears, it allows the user to select an image file from the desktop.

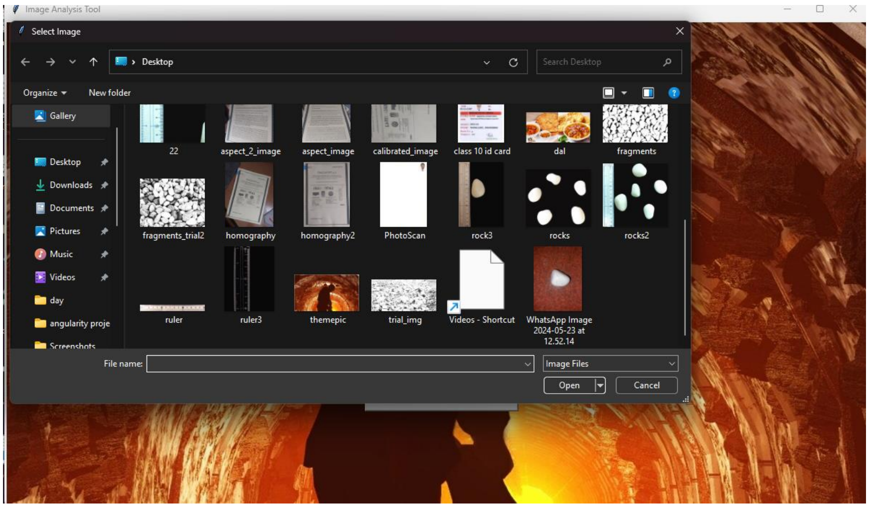


Figure 51 select the image

- **Step 2: cropping of selected image**

  After the image is selected, user then crops the image, this allows contour detection in selected region. After cropping the image when user presses “Enter” button, we the contoured image of the rock fragments, as shown in fig.

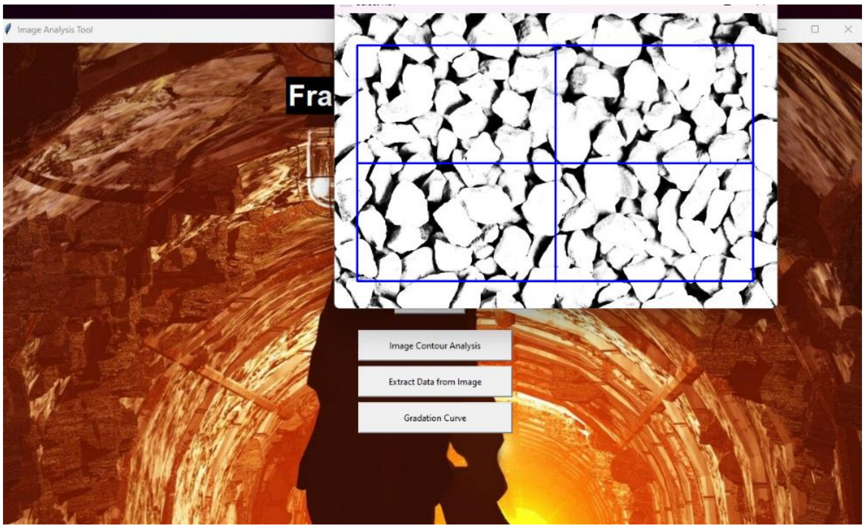


Figure 52 Cropping of the selected image

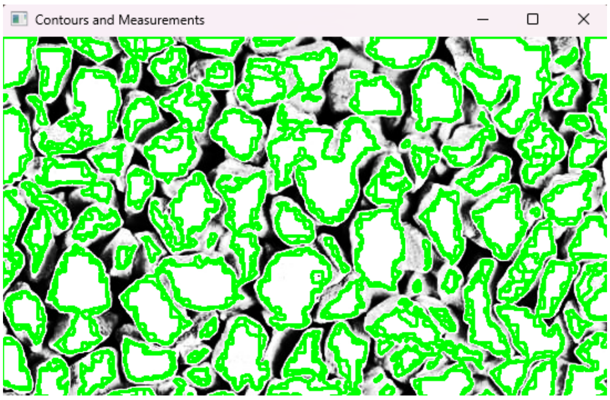


Figure 53 Contouring on the rock fragments

- **Step 3: User confirmation**
  After the Contours and Measurements window is closed, a dialog box appears asking if the user is satisfied with this contour detection, user either selects “Yes” or “No”
  If the user enters YES the image is then saved in the desktop.
  If the user enters NO the image then goes for manual contouring where the user manually contours the image.

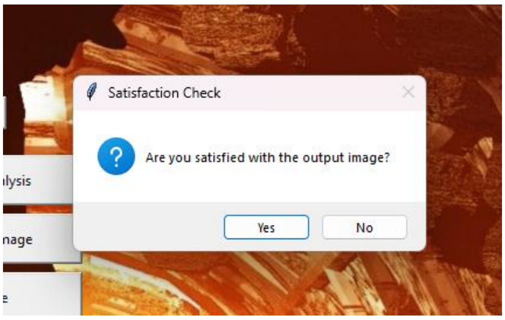


Figure 54 User satisfied or not

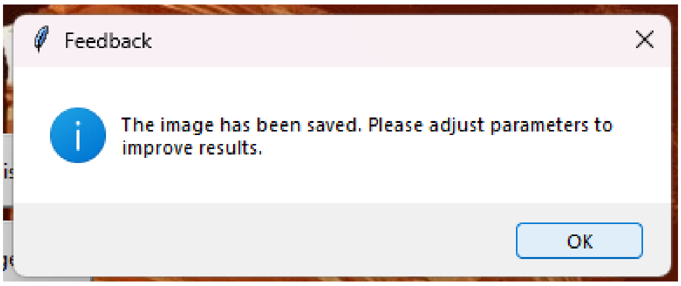


Figure 55 If no, image saved in desktop

- **Step 4: Manual Contouring**
  When user is not satisfied with the contouring, they can manually contour the image, user open the saved Contoured image from the desktop, then goes to edit, using red colour then can contour the region they want to analyse about.

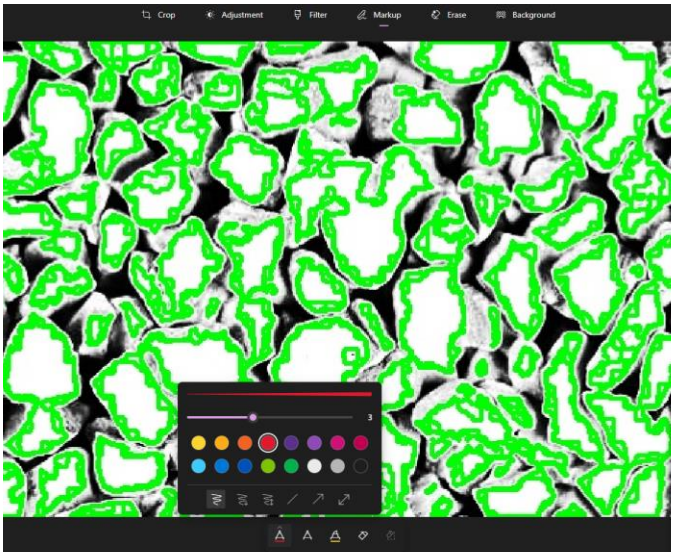


Figure 56 Manual Contouring

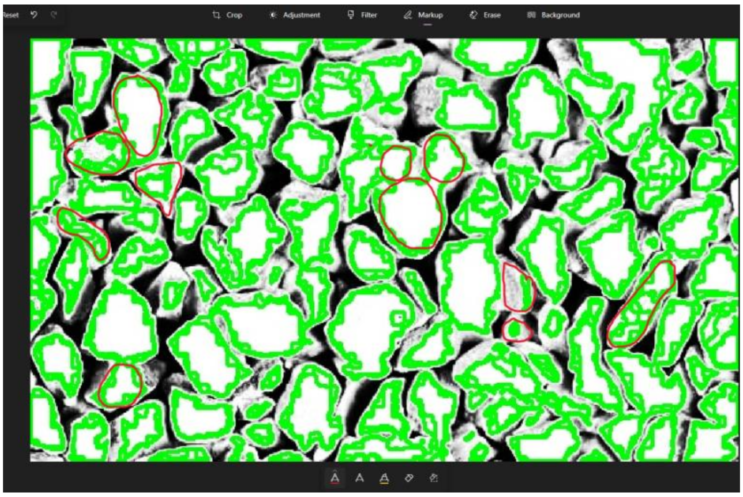


Figure 57 Red lines shows manual contouring

6.4.2 Extract Data from Image

- **Step 1 Selecting the contoured image**

  When the user clicks on "Extract Data from Image" a file dialog box appears, that enables the user to select the 'contoured_image' from the desktop.

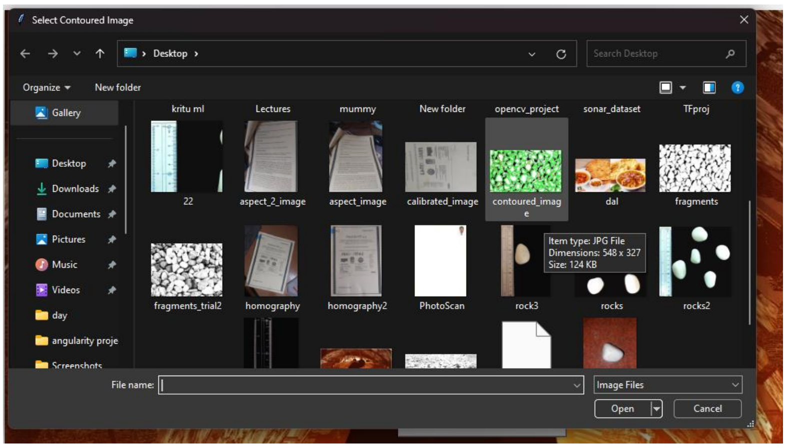


Figure 58 Selecting recently saved Contoured image

- **Step 2 User enters the density value**

  After the selection of image, a dialog box appears that asks the user to input the density value of the rock fragments, density value enables in the finding the value of mass and volume of the rock fragments that further helps in the analysis.

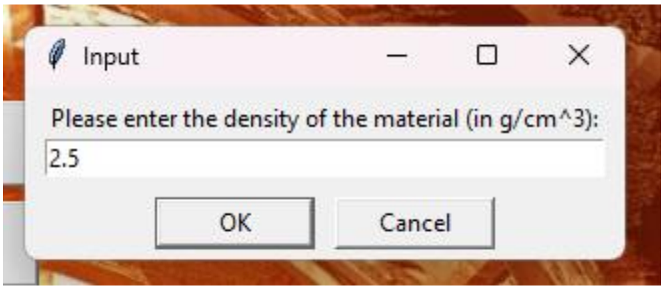
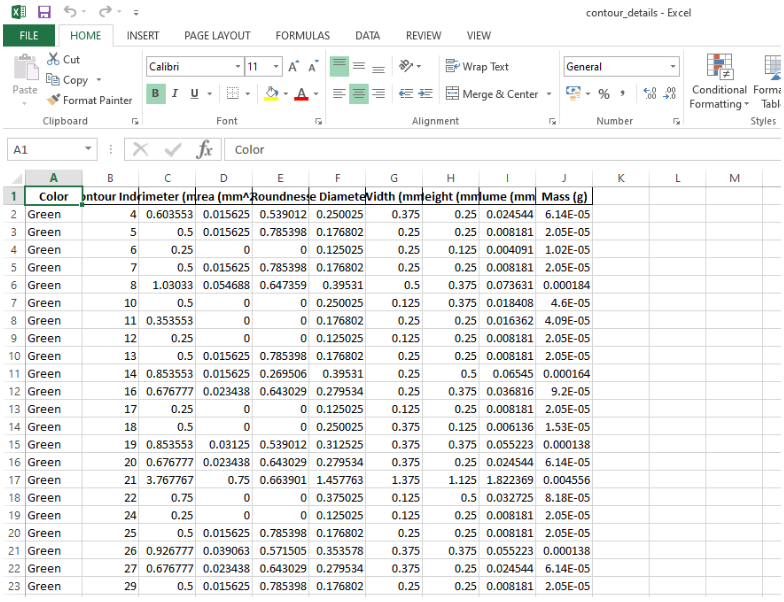


Figure 59 User enters density value of the rock

- **Step 3 Generating CSV File**

  The selected image is then processes and extracts the relevant data from the contoured image, like perimeter, diameter, roundness, height, width, mass and Volume.



	Color	ontour Ind	rimeter (m	rea (mm^	Roundness	e Diamete	Vidth (mm	eight (mm	lume (mm	Mass (g)
2	Green	4	0.603553	0.015625	0.539012	0.250025	0.375	0.25	0.024544	6.14E-05
3	Green	5	0.5	0.015625	0.785398	0.176802	0.25	0.25	0.008181	2.05E-05
4	Green	6	0.25	0	0	0.125025	0.25	0.125	0.004091	1.02E-05
5	Green	7	0.5	0.015625	0.785398	0.176802	0.25	0.25	0.008181	2.05E-05
6	Green	8	1.03033	0.054688	0.647359	0.39531	0.5	0.375	0.073631	0.000184
7	Green	10	0.5	0	0	0.250025	0.125	0.375	0.018408	4.6E-05
8	Green	11	0.353553	0	0	0.176802	0.25	0.25	0.016362	4.09E-05
9	Green	12	0.25	0	0	0.125025	0.125	0.25	0.008181	2.05E-05
10	Green	13	0.5	0.015625	0.785398	0.176802	0.25	0.25	0.008181	2.05E-05
11	Green	14	0.853553	0.015625	0.269506	0.39531	0.25	0.5	0.06545	0.000164
12	Green	16	0.676777	0.023438	0.643029	0.279534	0.25	0.375	0.036816	9.2E-05
13	Green	17	0.25	0	0	0.125025	0.125	0.25	0.008181	2.05E-05
14	Green	18	0.5	0	0	0.250025	0.375	0.125	0.006136	1.53E-05
15	Green	19	0.853553	0.03125	0.539012	0.312525	0.375	0.375	0.055223	0.000138
16	Green	20	0.676777	0.023438	0.643029	0.279534	0.375	0.25	0.024544	6.14E-05
17	Green	21	3.767767	0.75	0.663901	1.457763	1.375	1.125	1.822369	0.004556
18	Green	22	0.75	0	0	0.375025	0.125	0.5	0.032725	8.18E-05
19	Green	24	0.25	0	0	0.125025	0.125	0.25	0.008181	2.05E-05
20	Green	25	0.5	0.015625	0.785398	0.176802	0.25	0.25	0.008181	2.05E-05
21	Green	26	0.926777	0.039063	0.571505	0.353578	0.375	0.375	0.055223	0.000138
22	Green	27	0.676777	0.023438	0.643029	0.279534	0.375	0.25	0.024544	6.14E-05
23	Green	29	0.5	0.015625	0.785398	0.176802	0.25	0.25	0.008181	2.05E-05

Figure 60 Generating CSV file

6.4.3 Gradation Curve

- **Step 1 Selecting CSV file**

  When user clicks on “Gradation Curve” a file dialog box appears, user then selects the CSV file from the desktop.

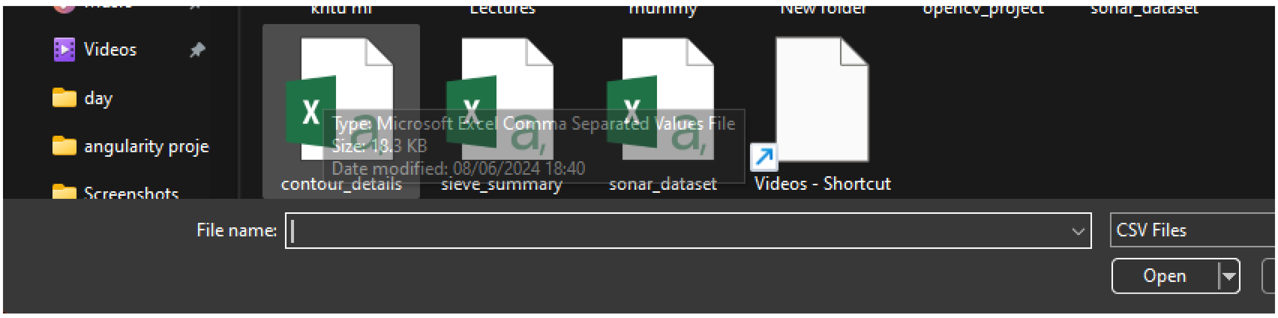


Figure 61 Selecting CSV file

- **Step 2 Generating Gradation Curve**

  The CSV file is processed to generate Gradation curve.

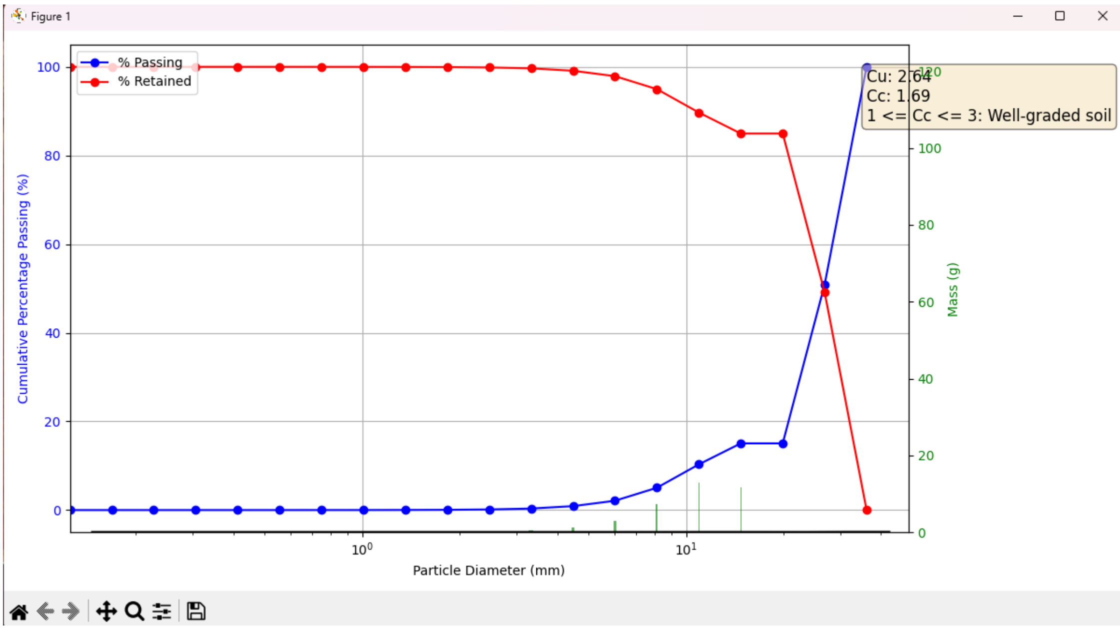


Figure 62 Final output, i.e. Gradation curve of the rock fragment analysis

CHAPTER 7

SUMMARY CONCLUSION AND FUTURE WORK

7.1 Summary

Background, motivation, importance of developing an image processing software, scope and objectives were emphasised in the beginning of the study. Existing methodology, limitations were highlighted in the literature review.

In Image processing various techniques were detailed that includes Grayscale Conversion, Gaussian Blurring, Histogram Equalization, Thresholding, and morphological operations. After image analysis we moved toward shape analysis that detailed us about contouring the image, Calibration process, calculation of area, volume and mass providing us a depth of theoretical and code implementation.

Sieve analysis principles were examined, which focused on the analysis and interpretation of results. This included determination of D-values, Coefficient of Uniformity and curvature, and lastly the generation of Gradation curve that helped in interpreting particle size distribution.

The technologies used in this project was mainly python libraries and GUI tools, the integration of these tools helped in image analysis, data handling and user interaction.

Lastly, a detailed implementation guide was provided that showcased software's functionality by step-by-step demonstration.

7.2 Conclusion

The project overall demonstrate the power of combining image processing with sieve analysis to offer a reliable tool for scientific and industrial applications. The software not only automates complex tasks but also ensures accuracy and consistency in result, making it an invaluable resource for researchers and practitioners.

7.3 Scope for future Study

While this project has developed a successful tool for image processing, there are several fields in the projects that need scope of improvement in future studies.

1. Enhanced image processing technique: Future work could explore more advanced image processing technique, such as advanced machine learning models for contour detection and classification, to improve the accuracy and efficiency.
2. 3D Volume calculation: Expanding the software to support 3D volume calculations using multiple image slices or 3D reconstruction techniques could provide better analysis of irregular shaped objects.
3. Real-Time Processing: Implementing real-time image processing capabilities would enable the software to analyse live video feeds, making it useful for dynamic applications such as quality control in manufacturing processes.

4. Mobile and Web Applications: Developing mobile and web-based versions of the software could increase its accessibility and convenience, allowing users to perform image analysis and sieve analysis on-the-go or via cloud-based platforms.

References

FG Carpenter, V. D. (1950). Methods of sieve analysis with particular reference to bone char.

G.H.A. Janaka J. Kumara, K. H. (2012). Image Analysis Techniques on Evaluation of Particle Size Distribution of Gravel. *International Journal of GEOMATE* .

Gedraite, E. S. (2011). Investigation on the effect of a Gaussian Blur in image filtering and segmentation. *IEEE*.

Ivan Culjak, D. A. (2012). A brief introduction to OpenCV. *IEEE*.

John M.Kemeny, A. D. (1993). Analysis of Rock Fragmentation Using Digital Image Processing.

Kennedy C. Onyelowe, J. S. (2021). Intelligent prediction of coefficients of curvature and uniformity of hybrid cement modified unsaturated soil with NQF inclusion.

Lei Shu, G. R. (2018). An automatic methodology for analyzing sorting level of rock particles.

M.Abdullah-Al-Wadud, M. H. (2007). A Dynamic Histogram Equalization for Image Contrast Enhancement. *IEEE*.

Mary L. Comer, E. J. (1999). Morphological operations for color image processing.

McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics.

Moore, A. (2018). *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter.*

Moruzzi, G. (2020). Plotting with Matplotlib. *SpringerLink*.

N.B. Rais, M. H. (2005). Adaptive thresholding technique for document image analysis. *IEEE*.

Saravanan, C. (2010). Color Image to Grayscale Image Conversion. *IEEE*.

Stefan van der Walt, S. C. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *IEEE*.

Ta Yang Goh, S. N. (2017). Performance analysis of image thresholding: Otsu technique.