A Project-II Report On

QUALITY ANALYSIS AND CLASSIFICATION OF RICE USING IMAGE PROCESSING

Submitted in partial fulfillment of the requirement For
The award of degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING 2018-2022

Submitted By

CH.UDHAY	18311A05R4
RISHABH SEMWAL	18311A05W2
KRISHNA GOEL	18311A05U3

Under the Guidance of
G.RAVI
Associate Professor

Department of Computer Science & Engineering
SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY
(AUTONOMOUS)
Yamnampet (V), Ghatkesar (M), Hyderabad – 501301, T.S.

AY-2021-2022

Department of Computer Science and Engineering

Sreenidhi Institute of Science and Technology

CERTIFICATE

This is to certify that this Project-II report on "**Quality Analysis and Classification of Rice using Image Processing**", submitted by **Ch.Udhay (18311A05R4) ,Rishabh Semwal (18311A05W2), Krishna Goel (18311A05U3)** in the year 2022 in the partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide work that has been carried out by them as a part of their **Project-II during Fourth Year Second Semester,** under our guidance. This report has not been submitted to any other Institute or university for the award of any degree.

Internal guide	**Project Coordinator**	**Head of the Department**
G.Ravi	Ms. N.Shivani	Dr. Aruna Varanasi
Associate Professor	Assistant Professor	Professor & HOD
Department of CSE	Department of CSE	Department of CSE

Signature of the External Examiner

Date:-

DECLARATION

We **Ch.Udhay (18311A05R4), Rishabh Semwal (18311A05W2) , krishna Goel (18311A05U3)** students of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR,** studying IV year II semester, **COMPUTER SCIENCE AND ENGINEERING** solemnly declare that the Project-II work, titled "**Quality Analysis and Classification of Rice using Image Processing"** is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** for partial fulfillment for the award of degree of Bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

ACKNOWLEDGEMENT

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **DR. T. Ch. Siva Reddy,** who most ably runs the institution and has had the major hand in enabling me to do my project.

I profoundly thank **Dr. Aruna Varanasi**, Head of the Department of Computer Science & Engineering who has been an excellent guide and also a great source of inspiration to my work.

I would like to thank our Coordinator **Ms. N. Shivani** & my internal guide **G.Ravi** for Project-
II for their technical guidance, constant guidelines, encouragement and support in carrying out my project on time at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

Ch.Udhay	**18311A05R4**
Rishabh Semwal	**18311A05W2**
Krishna Goel	**18311A05U3**

ABSTRACT

Grain quality analysis is a huge challenge in agricultural industries. Internal control is critical in the food industry because food products are characterized and rated into various categories after quality data has been collected. Grain quality assessment is performed by hand, but the results are subjective, lengthy, and pricey. To overcome the limitations and drawbacks of image processing techniques, different resolutions are used for grain quality analysis. Using image processing techniques, this paper proposes a method for grading and analyzing rice based on grain size and form. An edge detection algorithmic software is used in particular to determine the area of each grain's borders. we discover the endpoints of each grain using this technique.

LIST OF FIGURES			
S.NO	**Figure No.**	**Title of Figure**	**Page No**
1	**4.1**	**System Architecture**	**8**
2	**4.2**	**Data Flow Diagram**	**8**
3	**4.3.1**	**Class Diagram**	**10**
4	**4.3.2**	**Use case Diagram**	**10**
5	**4.3.3**	**Sequence Diagram**	**11**
6	**4.3.4**	**Activity Diagram**	**12**

INDEX

	Page No
List of Figures	**vi**
1. INTRODUCTION	**1**
1.1 Project Introduction	**1**
1.2 Scope	**2**
1.3 Project Overview	**2**
1.4 Objectives	**3**
2. LITERATURE SURVEY	**3**
2.1 Existing System	**3**
2.2 Proposed System	**3**
2.3 Related Work	**4**
3. SYSTEM ANALYSIS	**5**
3.1 Functional Requirements	**5**
3.2 Performance Requirements	**6**
3.3 Software Requirements	**6**
3.4 Hardware Requirements	**6**
4. SYSTEM DESIGN	**7**
4.1 System Architecture	**8**
4.2 Data Flow Diagram	**8**
4.3 UML Diagrams	**9**
4.3.1 Class Diagram	**9**
4.3.2 Use Case Diagram	**10**
4.3.3 Sequence Diagram	**11**
4.3.4 Activity Diagram	**12**
5. IMPLIMENTATION AND RESUTS	**13**
5.1 Language / Technology Used	**13**
5.2 Methods / Algorithms Used	**14**
5.3 Sample Code	**15**
5.4 Results (Accuracy) / Output Screens	**31**
6. TESTING	**32**
6.1 Types of Testing	**32**
6.2 Test Cases	**33**
7. CONCLUSION	**36**

8. FUTURE SCOPE	**37**
9. BIBLIOGRAPHY	**37**
APPENDIX-A :	**38**
PLAGIARISM REPORT	**50**
PAPER PUBLICATION	**51**
ABSTRACT	**67**
PROJECT CORRELATION WITH PO’s AND PSO’s	**68**
NATURE OF THE PROJECT	**69**
DOMAIN OF THE PROJECT	**70**

1. INTRODUCTION

1.1 Project Introduction

The agriculture enterprise is the oldest and maximum huge enterprise withinside the world. Traditionally, the first-class of meals are described via way of means of a human sensory panel primarily based totally on its bodily and chemical properties. Physical parameters encompass grain length and form, moisture content material, chalk, whiteness, freeness, and bulk density. Moisture content material is not anything extra than the water content material in the grain. For the top-of-line storage, the moisture content material needed to be among 12-14%. Various techniques are used for moisture evaluation.

The widely used moisture meter and the convection oven method. Chalk is a white spot found within the endosperm of rice. Calcareous grain is half of the grain that is white in color and brittle in nature. Calcareous grains decompose at some point during milling due to their brittleness, and this has an effect on the rice's milling diploma. Rice grains are classified into three types based on chalk: white belly, white center, and white back. Chalk rice reduces the palatability of cooking products, thereby reducing the presence of chalk by more than 20% on the market. For chalk detection, use a magnifying glass and a photographic magnifying glass. The paper focused on grain length and form evaluation of image processing strategies. Grain length and form are measured using a pointer micrometer, a graphical approach, and a grain form tester. However, these techniques take time and a number of them are costly. To overcome these constraints, image processing strategies provide an opportunity and a highly satisfactory solution.

The proposed method's main goal is to provide an alternative solution for quality analysis that reduces the time and cost required. Image processing is a vital and advanced technological field that has seen significant advancements. Attempts are being made to replace the traditional human sensory panel. The document is a solution to agribusiness problems.

1.2 Scope

For the detailed analysis of quality products related to accuracy and efficiency performance, various types of grain and their varieties are used. Instead of a scanner, we can use a conveyor belt with a vibration mechanism for the experiment to use the appropriate hardware. The collection, evaluation, protection, and trade of rice germplasm, as well as the distribution of improved plant material to various national and local research centers. Development of innovation for incorporated nuisance and illness, as well as supplementing the board for various farming circumstances.

The majority of quality analysis factors must be measured using image processing techniques. This research could be expanded to develop a method for identifying granules based on any attribute that can be used to improve rice quality. The cost of such a system should be low, as should the time spent on quality analysis.

1.3 Project Overview

We use digital image processing, enhancement, and analysis in this project to assess the quality of various rice samples. The image is spatially processed. To determine the size, color, and overall quality of rice samples, image reduction, image enhancement, and image increment object recognition in the spatial domain are applied grain by grain. Grain quality is assessed manually, but it is subjective, time-consuming, and can lead to costly variations. The rice grains are evaluated based on their basic size and shape to find the region of boundaries in each grain using an image processing edge detection algorithm. We measure the length and breadth of rice grains after determining the endpoints of each grain. The performance of image processing reduces operation time.

Many studies have been conducted to determine the best methodology for segmenting rice kernels that are touching and/or overlapping. They experimented with rice grain classification using machine vision and machine learning algorithms, obtaining rice image types spread out on a solid black platform using a web camera. During the pre-processing stage of their work, they apply filters to gray scaled images. Erode and dilate operations were used to separate grain images that were touching. Each grain image is extracted using contour segmentation. It used a camera with a black background, uniform illumination, and a constant distance between the camera and the rice sample to photograph rice grains. They segment using adaptive thresholding. To detect edges, Canny edge detection is used.

1.4 Objectives

The proposed method's primary goal is to provide an alternative solution for quality analysis that reduces the required time and cost. Image processing is a vital and advanced technological field that has seen significant advancement. Attempts are underway to replace the traditional human sensory panel.

2. LITERATURE SURVEY

2.1 Existing System

Traditionally, the quality of a food product is defined by its physical and chemical properties by a human sensory panel, which is time consuming, can produce inconsistent results, and is expensive. Traditional human sensory panel, because human operations are inefficient and inconsistent.

Frame works like R, Numpy, pandas exist to pre-process the data using the environments like Rstudio, Jupyter notebook, anaconda which requires programming skills and frame work knowledge.

2.2 Proposed System

The image processing technique counts the number of rice seeds and categorizes them based on length, breadth, and length – breadth ratio. The average length of rice grain is length, while the average breadth of rice grain is breadth, and the length-breadth ratio is calculated as:

L/B = [(Average length of rice grain)/(average breadth of rice)]*100

In the proposed system data is passed through 5 stages, where we perform different operations to achieve accuracy.

• Image registration occurs in the first pre-processing step, and noise is removed from the image using a filter.

• The shrinkage algorithm is used in the second step to segment the touching kernels.

• In the third step, we perform edge detection to determine the boundary region.

• In the fourth step, rice seed measurement is performed, as well as length, breadth, and length-breadth measurement.

• Rice is classified in the fifth step of the algorithm based on its size and shape.

2.3 Related Work

Many studies have been conducted in order to determine the best methodology for segmenting rice kernels that are touching and/or overlapping. They tested rice grain classification using machine vision and machine learning algorithms, capturing images of all rice granules spread out on a solid black platform with a web camera. To pre-process a greyscale image, they used a Gaussian filter with a 3x3 kernel size, as well as pyramid down and pyramid up filters. Erode and dilate operations were used to separate grain images that were touching. Each grain image was extracted using contour segmentation.

An analysis method for rice granules that employs computer vision techniques enabled by an Artificial Neural Network (ANN). To capture an image of the grains, it used a CCD camera with a black background, uniform illumination, and a constant distance between the camera and the rice sample. They segment using adaptive thresholding. To detect edges, Sobel and Canny edge detection are used. To correct segmentation errors caused by non-uniform illumination, Top Hat Transformation was used. By providing uniform illumination with no dark or light areas, this method separated the foreground from the background. Objects and image edges are identified using threshold methods.

They proposed examining the appearance of rice using image processing. The shape and chalkiness of the rice kernels were used as parameters. The contacting angle method was used to segment the touching and overlapping rice seeds. They proposed a method for separating partially overlapping objects in silhouette images. The approximate shapes of the objects are known. They employed an edge-based segmentation technique. It was broken down into three steps: extraction of seed points, extraction of contour evidence, and estimation of contours.

They proposed using image processing to investigate the appearance of rice. As parameters, the shape and chalkiness of the rice kernels were used. To segment the touching and overlapping rice seeds, the contacting angle method was used. They proposed a technique for distinguishing partially overlapping objects in silhouette images. The objects' approximate shapes are known. They used an edge-based segmentation method. It was divided into three steps: seed point extraction, contour evidence extraction, and contour estimation.

3.SYSTEM ANALYSIS

This System Analysis resembles a requirements analysis. It is also defined as "an explicit formal inquiry conducted to assist someone in identifying a better course of action and making a better decision than he would have made otherwise." This step entails disassembling the system to analyze the situation, analyzing project goals, disassembling what needs to be created, and attempting to engage users in order to define definite requirements.

3.1 Functional Requirements

Functional requirements are the qualities or functions of a product. The functional requirements in our product are

3.1.1 Data extraction

Importing the libraries and requirements, training the models with the dataset containing images (real and synthetic) and create the code to test the images for the restoration.

3.1.2 Image processing

Image processing enables us to transform and manipulate a lot of images at once while extracting useful insights. It has numerous applications in almost every field. Python is a popular programming language for this purpose.

3.1.3 Dash overview

Dash is a user framework for developing interactive analytical web applications in pure Python or

R, built on Flask, Plotly.js, and React.js. It comes with hundreds of charts, graphs, and user interface controls, allowing you to build highly customized analytic apps with just a few lines of code.

3.1.4 Results evaluation

Provide the input images and evaluate the results of quality analysis and classification of rice grains.

3.2 Performance Requirements

The output of the application is used to assess performance. The specification of requirements is critical in system analysis. A system can only be designed to fit into the required environment if the requirement specifications are properly provided. Because they will eventually use the system, it is largely up to the existing system's users to provide the requirement specifications. This is because the requirements must be known early on so that the system can be designed appropriately. Once a system has been designed, it is extremely difficult to change it, and designing a system that does not meet the needs of the user is pointless.

3.3 Software Requirements

Operating System: Windows / Mac / Linux
Language: Python 3.7
Google (or any browser)
Text editor: Jupyter Notebook.

3.4 Hardware Requirements

Processor: Intel I5
RAM :8GB
Hard disk: 256 GB SSD drive

Processor	**:**	**Intel**
RAM	:	8GB
Hard disk	:	256GB

4.SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

This section provides a non-technical narrative description of the system. It should include a high-level system architecture diagram and, if applicable, a subsystem breakdown. If applicable, interfaces to external systems should be depicted in the high-level system architecture or subsystem diagrams. Provide a high-level context diagram for the system and subsystems, if applicable. Consult the requirements trace ability matrix (RTM) in the Functional Requirements Document to determine the allocation of functional requirements into this design document (FRD).

This section describes any system design constraints (such as resource use versus productivity or conflicts with other systems) and includes any assumptions made by the project team while developing the system design.

The association code and title of the central issues of contact (and substitutes if suitable) for the data framework advancement exertion.

4.1 System Architecture

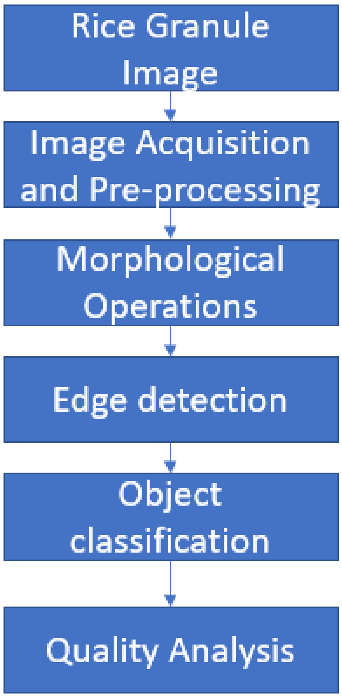


Figure 4.1 System Architecture

4.2 Data Flow Diagram

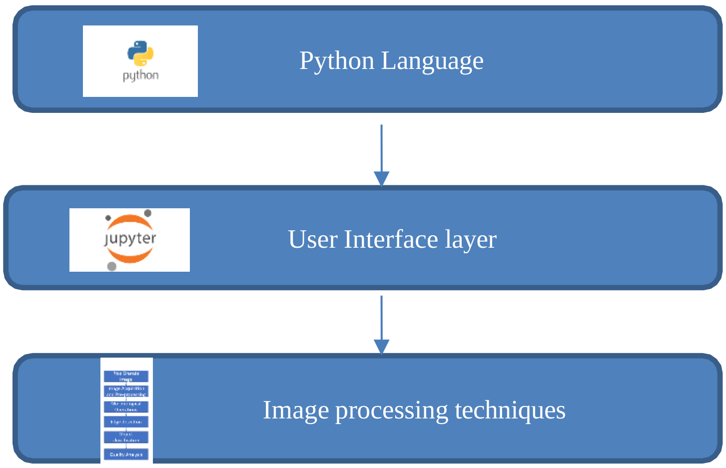


Figure 4.2 Data Flow Diagram

4.3 UML Diagrams

UML (Unified Modeling Language) is a common language for selecting, visualizing, creating, and specifying programming structures. It is in like manner used to display nonprogramming structures correspondingly like cycle stream in a get-together unit, etc.

UML has an instigated relationship with question-created assessment and blueprint. UML expect a central part in depicting exchange perspectives of a construction.

4.3.1 Class Diagram:

The class chart is the most typically pulled in format UML. It tends to the static strategy point of view of the construction. It cements the technique of classes, interfaces, joint endeavors, and their affiliations.

A Class Diagram in Software engineering is a static structure that gives an overview of a software system by displaying classes, attributes, operations, and their relationships between each other. This Diagram includes the class name, attributes, and operation in separate designated compartments. Class Diagram helps construct the code for the software application development. Class Diagram defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modeling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

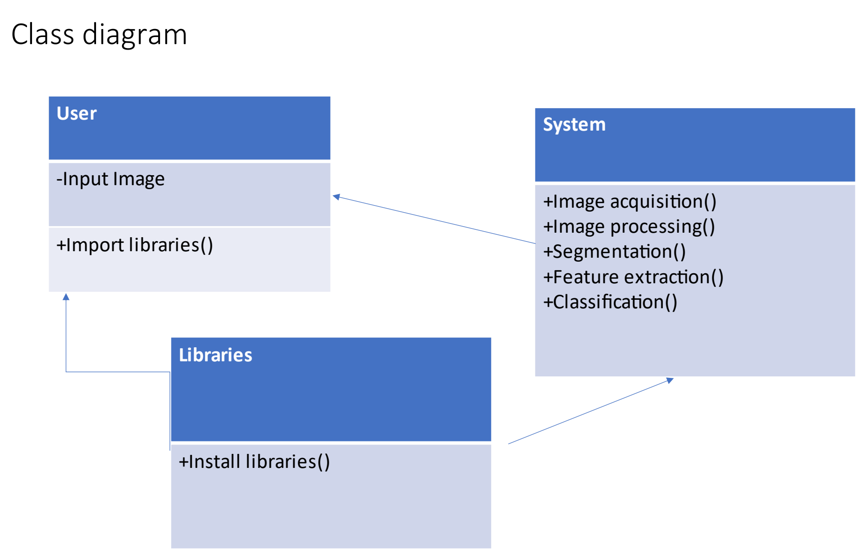


Fig: 4.3.1 Class Diagram

In the above class graph, the relationship that is the reliance between every last one of the classes is portrayed. Furthermore, even the tasks acted in every single class are comparably showed up.

4.3.2 Use case Diagram:

The use case diagram is for showing the direction of the construction. This diagram contains the game-plan of purpose cases, performing aces, and their relationship. This diagram may be used to address the static viewpoint of the construction.

The purpose of a Use Case Diagram is to give a graphical overview of the functionalities provided by a system in terms of actors, their goals (represented as use cases), and any dependencies among those use cases.

A Use Case Diagram describes the usage of a system. The associations between actors and use cases represent the communications that occur between the actors and the subjects to accomplish the functionalities associated with the use cases. The subject of a use case can be represented through a system boundary. The use cases enclosed in the system boundary represent the functionalities performed by behaviors (activity diagrams, sequence diagrams, and state machine diagrams).

Actors may interact either directly or indirectly with the system. They are often specialized so as to represent a taxonomy of user types or external systems. The only relationship allowed between actors in a use case diagram is generalization. This is useful in defining overlapping roles between actors. Actors are connected to use cases through communication paths, each represented by a relationship.

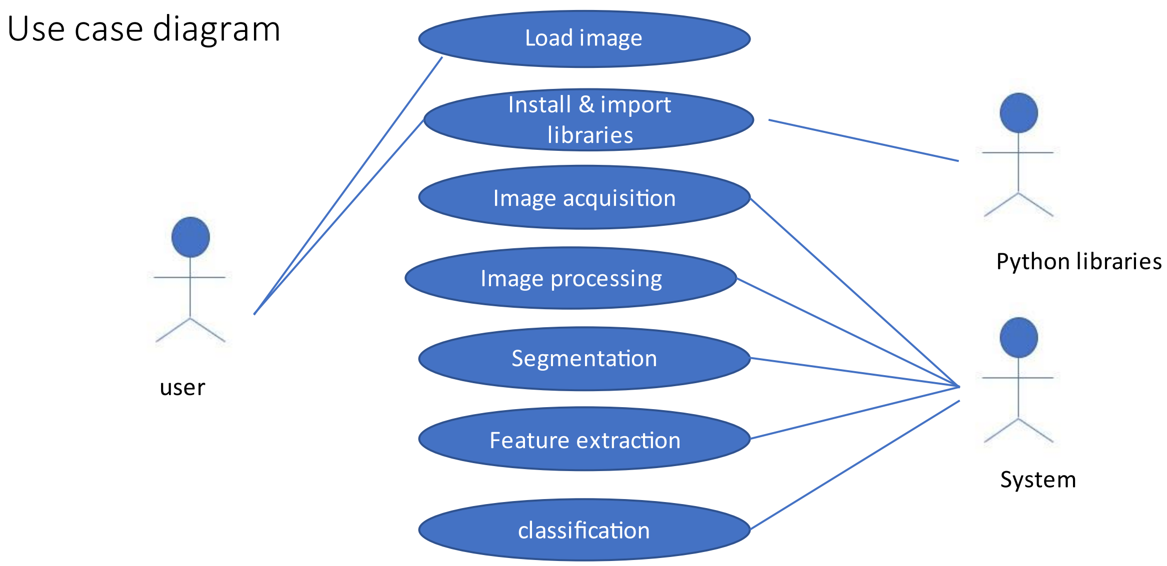


Fig: 4.3.2 Use Case Diagram

In the above diagram, the actor python libraries install and import required libraries, which are useful for extracting features by user, next the user load the input image and import required libraries and then system fits the input image into acquisition, Pre-processing, Segmentation, Feature extraction, Classification and finally evaluate the results.

4.3.3 Sequence Diagram

This is a participation plan which watches out for the time mentioning of messages, this diagram is used to address the powerful point of view of the design.

A progression frame shows question correspondences engineered in the time plan. In the underneath chart, three articles are helping out one another. Each dissent has an upward run line that addresses the presence of an inquiry throughout some vague period. This diagram has furthermore a tall, dainty square shape which is called the focal point of control that exhibits the period in the midst of which dissent is playing out an action, either explicitly or through a subordinate framework

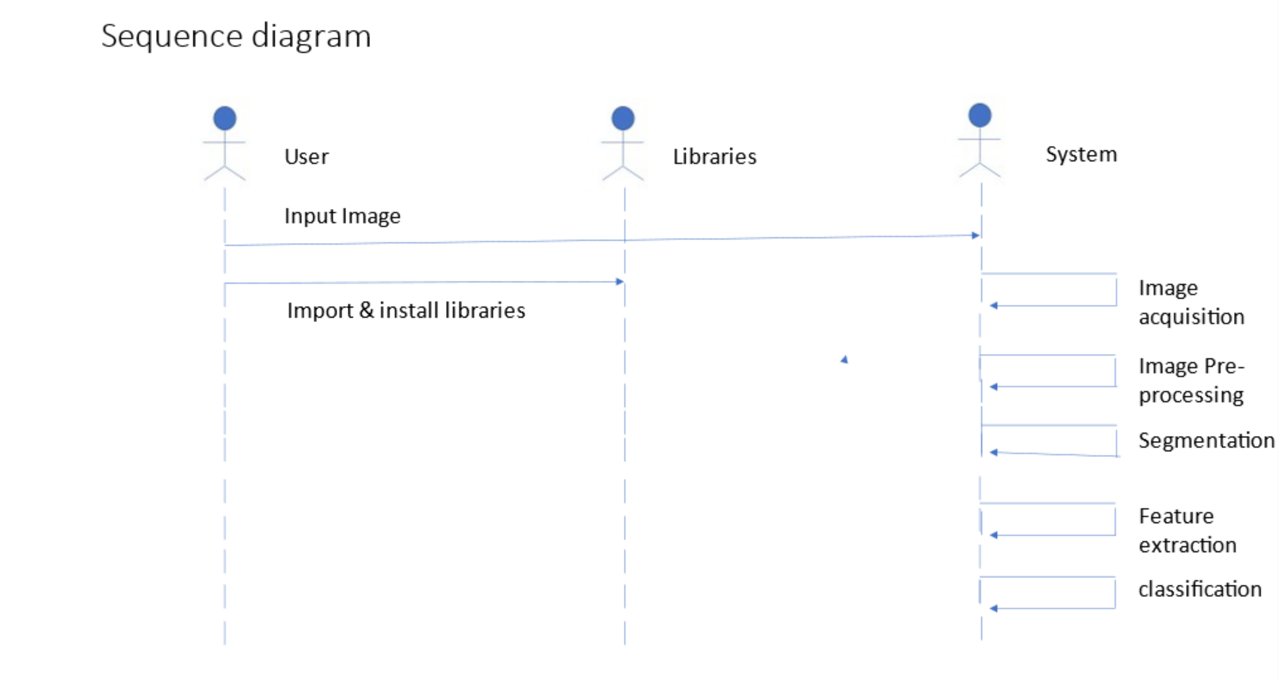


Fig: 4.3.3 Sequence Diagram

4.3.4 Activity Diagram:

An Activity in the movement chart is otherwise called an Activity state. It is utilized to address the summon of activity, a stage in a whole business process.

Swim lanes are utilized to show which exercises are performed by which association in the movement chart. The paths are limits are drawn and the exercises of a specific association are attracted to the very path as that of the association. Swim lanes must be arranged Logically. It is recommended to have under five swim lanes in an action chart. Swim lanes are great in that they consolidate the movement chart's portrayal of rationale with the connection graph's portrayal of obligation.

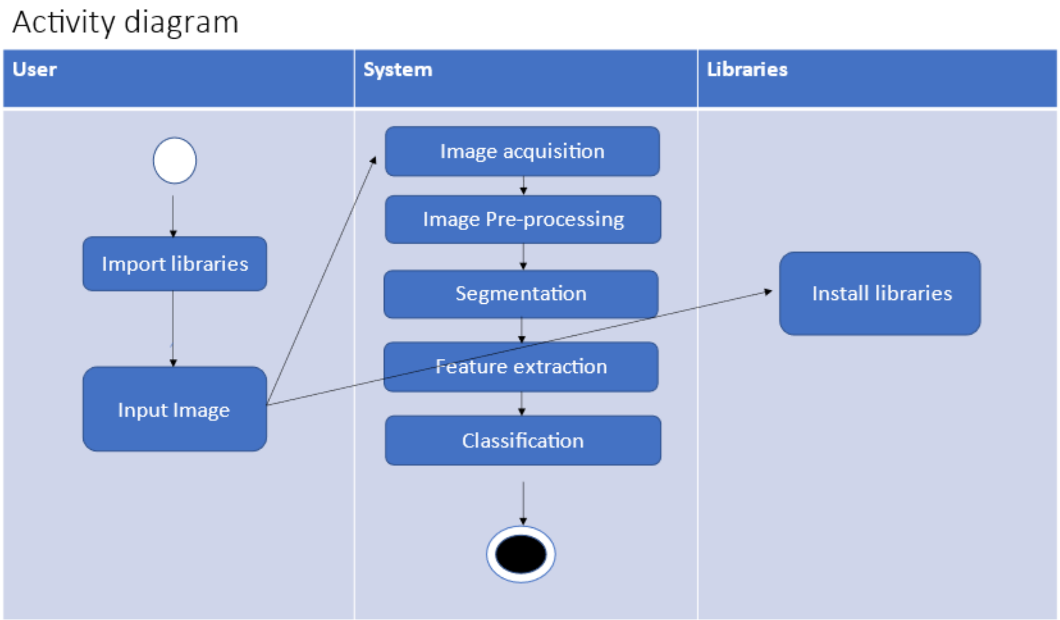


Fig: 4.3.4Activity Diagram

5. IMPLEMENTATION AND RESULTS

Any project's implementation stage is a true portrayal of the defining moments that decide whether or not a project succeeds. During the implementation stage, the system or system modifications are installed and rendered operational in a production environment. The phase begins once the system has been tested and accepted by the user. This phase will last until the system is in production and meets the needs of the users.

5.1 Language / Technology Used

Python programming language and Deep Learning technology are used in this project, the execution phase of any undertaking is a genuine presentation of the vital crossroads that make a task a triumph or a disappointment. The executionstage is characterized as the framework or framework changes being introduced and made functional in a creative climate. The stage begins once the client has reviewed and approved the framework. This stage continues till the structure is up and running in accordance with the defined customer requirements.

5.2 Methods/Algorithms used

5.2.1 Image pre-processing

To reduce noise that arises during image acquisition, a filter is utilised. The filter sharpens the image as well. The threshold technique is used to isolate the rice grains from the black backdrop.

5.2.2 Shrinkage morphological operation

Erosion is a technique for separating the touching features of rice grains while maintaining the integrity of each feature. The dilation process follows the erosion process. The purpose of dilation is to return eroded features to their former shape without having to re-join the separated components.

5.2.3 Edge detection

The process of detecting the region of rice grain boundaries is aided by edge detection. Six edge detection methods are available in the vision and motion toolbox: differentiation, gradient, Prewitt, Roberts, sigma, Sobel, and Canny. The edge detection filter that will be utilised is specified by the technique. To detect edges, we applied a clever algorithm in the proposed methodology.

5.2.4 Contours Approximation Method

Contours are the boundaries of a shape with the same intensity. It stores the (x, y) coordinates of the boundary of a shape. But does it store all the coordinates? That is specified by this contour approximation method. If we pass cv2.CHAIN_APPROX_NONE, all the boundary points are stored. But actually, do we need all the points? For eg, if we have to find the contour of a straight line. We need just two endpoints of that line. This is what cv2.CHAIN_APPROX_SIMPLE does. It removes all redundant points and compresses the contour, thereby saving memory.

5.2.5 Object measurement

Measurement is used to determine the number of rice grains. Edge detection methods are applied to the image once the number of rice grains is determined, producing endpoint values for each grain. To link the terminals and measure the length and breadth of each grain, calliper is utilized. Once we have the length and breadth numbers from contours of cv2.boundingRect() function, we can calculate the length-breadth ratio or aspect ratio.

5.2.6 Object classification

All standard, measured, and calculated findings must be used in classification. The Directorate of Rice Research, Rajendra Nagar, Hyderabad, has a laboratory handbook on rice grain quality that serves as the standard database for rice size and shape measurement.

The following tables show the classification of rice grains according to the standard database. The table below shows rice grain classification based on length and length-breadth ratio:

SLENDER	Aspect ratio>=3 and aspect ratio<3.5
MEDIUM	Aspect ratio>=2.1 and aspect ratio<3
BOLD	Aspect ratio>=1.1 and aspect ratio<2.1
ROUND	Aspect ratio>=0.9 and aspect ratio<1
DUST	Aspect ratio>3.5

5.3 Sample code

5.3.1 Import the essential libraries

NumPy

NumPy is a Python module that adds support for huge, multi-dimensional arrays and matrices, as well as a vast library of high-level mathematical functions to work on them.

matplotlib

Matplotlib is a Python 2D plotting toolkit that creates high-quality figures in a range of hardcopy and interactive formats on a variety of platforms. Plots, histograms, power spectra, bar charts, error charts, scatterplots, and other visuals can all be created with Matplotlib.

Pandas

Pandas is a library for data manipulation and analysis. The library provides data structures and methods for managing numerical tables and time series.

Dash

Dash is a Python framework for building analytical web applications that is free and open-source. It's a robust library that simplifies the creation of data-driven applications. It's especially useful for Python data scientists who aren't used to working with the web. Dash lets users create beautiful dashboards right in their browser. Dash integrates modern UI components like dropdowns, sliders, and graphs to your analytical Python code using Plotly.js, React, and Flask. Dash apps are made up of a Flask server that sends HTTP requests with JSON packets to front-end React components. Dash applications do not require HTML or JavaScript because they are created purely in Python.

Plotly

Plotly is a Montreal-based technical computer firm that makes data analytics and visualisation solutions including Dash and Chart Studio. It has developed open-source graphing API libraries for Python, R, MATLAB, Java script, and other programming languages.

The following are some of Plotly's key features:

• It generates interactive graphs.

• The graphs are saved in JavaScript Object Notation (JSON) data format, which allows them to be read by scripts written in other programming languages such as R, Julia, MATLAB, and so on.

• Graphs can be exported in a variety of raster and vector image formats.

OpenCV-Python

The Python binding library OpenCV-Python is used to solve computer vision challenges. Use the cv2.imshow() method to show an image in a window. The image's size is automatically adjusted in the window.

Base64

Base64 is a group of binary-to-text encoding techniques that convert binary data (particularly, a sequence of 8-bit bytes) into 24-bit sequences represented by four 6-bit Base64 digits.

Base64, like all binary-to-text encoding techniques, is designed to transfer binary data over networks that can only reliably carry text content. Base64 is particularly common on the Internet, where it is used to embed picture files and other binary assets within textual assets like HTML and CSS files.

```
In [1]: import dash
from dash import dcc
from dash import html
import plotly.graph_objects as go
from plotly import subplots
import pandas as pd
import plotly.express as px
import numpy as np
from dash.dependencies import Input, Output, State
import cv2
import PIL.Image as image
from io import BytesIO
import base64
from matplotlib import pyplot as plt
app = dash.Dash(__name__)
```

5.3.2 Classification of rice particles

```
In [2]: def get_classification(ratio):
            ratio =round(ratio,1)
            toret=""
            if(ratio>=3 and ratio<3.5):
                toret="Slender"
            elif(ratio>=2.1 and ratio<3):
                toret="Medium"
            elif(ratio>=1.1 and ratio<2.1):
                toret="Bold"
            elif(ratio>0.9 and ratio<=1):
                toret="Round"
            else:
                toret="Dust"
            return toret
```

5.3.3 Initialising Values

```
In [3]: classification = {"Slender":0, "Medium":0, "Bold":0, "Round":0, "Dust":0}
        avg = {"Slender":0, "Medium":0, "Bold":0, "Round":0, "Dust":0}
```

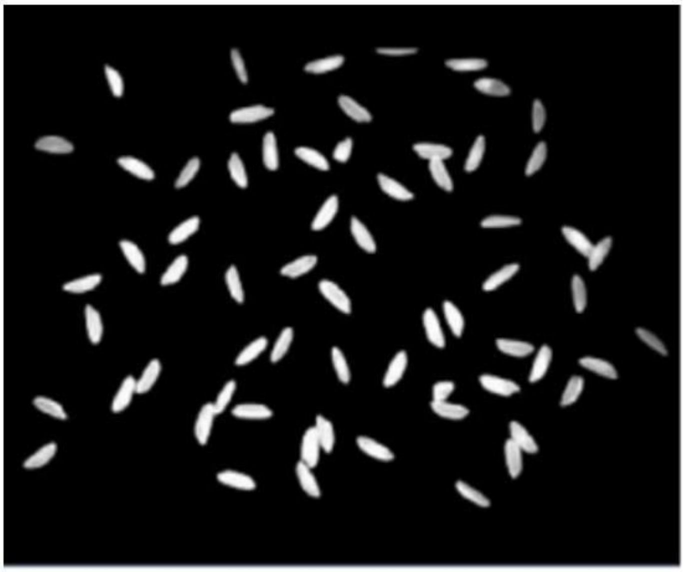
5.3.4 Load in grayscale mode

Grayscaling is the process of transforming a picture from RGB, CMYK, HSV, and other colour schemes to shades of grey. It could be entirely black or entirely white. Grayscaling and its importance.

・ Dimension reduction: For example, RGB images have three colour channels and three dimensions, whereas grayscale images have only one.

・Lowers model complexity: Consider training neural articles on 10x10x3 pixel RGB images. There will be 300 input nodes in the input layer. However, for grayscale images, the same neural network will only require 100 input nodes.

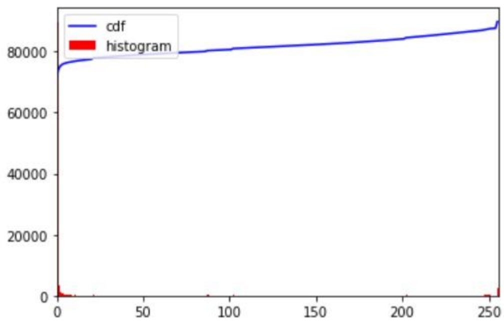
In [4]:
```
from IPython.display import display, Image
img = cv2.imread('C:/Users/lenevo/Downloads/rice2.png',0)
display(Image(filename='C:/Users/lenevo/Downloads/rice2.png'))
```



5.3.5 Histogram part of the image

A histogram is a type of graph. A graph that displays the frequency of anything. Histograms typically have bars that represent the frequency of occurrence of data across the entire data set.

In [5]:
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('C:/Users/lenevo/Downloads/rice2.png',0)

hist,bins = np.histogram(img.flatten(),256,[0,256])

cdf = hist.cumsum()
cdf_normalized = cdf * hist.max()/ cdf.max()

plt.plot(cdf_normalized, color = 'b')
plt.hist(img.flatten(),256,[0,256], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()
```



5.3.6 Convert into Binary

A binary picture is a monochromatic image composed of pixels that can only have one of two colors: black or white. Bi-level or two-level images are other terms for binary images. Each pixel is thus represented by a single bit, either 0 or 1.

```
In [6]: # 160 - threshold, 255 - value to assign, THRESH_BINARY_INV - Inverse binary
        ret,binary = cv2.threshold(img,160,255,cv2.THRESH_BINARY)
```

5.3.7 Averaging Filter

The Averaging Filter technique replaces the central element with the average of all pixels under the kernel area.

```
In [7]: kernel = np.ones((5,5),np.float32)/9
        dst = cv2.filter2D(binary,-1,kernel)
        # -1 : depth of the destination image
        kernel2 = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3))
```

5.3.8 Erosion

```
In [8]: erosion = cv2.erode(dst,kernel2,iterations = 1)
```

5.3.9 Dilation

```
In [9]: dilation = cv2.dilate(erosion,kernel2,iterations = 1)
```

5.3.10 Edge detection

```
In [10]: edges = cv2.Canny(dilation,100,200)
```

5.3.11 Size detection

Contours Approximation Method

We can see that contours are the same-intensity boundaries of a shape. It saves the (x, y) coordinates of a shape's boundary. But does it keep track of all the coordinates? This contour approximation method specifies it. All boundary points are captured if we use cv2.CHAIN APPROX NONE. But do we really need all of the points? For example, suppose we need to find the contour of a straight line. We only need two endpoints of that line. In that case, cv2.CHAIN APPROX SIMPLE can be used. It removes all unnecessary points and compresses the contour, saving memory.

```
In [11]: contours, hierarchy = cv2.findContours(erosion, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
         print("No. of rice grains=",len(contours))
         total_ar=0

         No. of rice grains= 52
```

5.3.12 Counting impurities

```
In [12]: for cnt in contours:
             x,y,w,h = cv2.boundingRect(cnt)
             aspect_ratio = float(w)/h
             if(aspect_ratio<1):
                 aspect_ratio=1/aspect_ratio
             #print(round(aspect_ratio,2),get_classification(aspect_ratio))
             classification[get_classification(aspect_ratio)] += 1
             if get_classification(aspect_ratio) != "Dust":
                 total_ar+=aspect_ratio
             if get_classification(aspect_ratio) != "Dust":
                 avg[get_classification(aspect_ratio)] += aspect_ratio
```

5.3.13 Getting average value

```
In [13]: avg_ar=total_ar/len(contours)
```

5.3.14 Setting values for Classification of rice

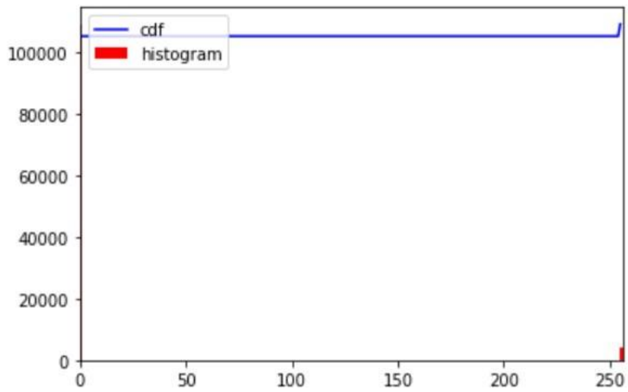
```
In [14]: if classification['Slender']!=0:
             avg['Slender'] = avg['Slender']/classification['Slender']
         if classification['Medium']!=0:
             avg['Medium'] = avg['Medium']/classification['Medium']
         if classification['Bold']!=0:
             avg['Bold'] = avg['Bold']/classification['Bold']
         if classification['Round']!=0:
             avg['Round'] = avg['Round']/classification['Round']
```

5.3.15 Saving different types of images

```
In [15]: cv2.imwrite('C:/Users/lenevo/Downloads/rice2.png', img)
         cv2.imwrite('C:/Users/lenevo/Downloads/Binary.png', binary)
         cv2.imwrite('C:/Users/lenevo/Downloads/dust.png', dst)
         cv2.imwrite('C:/Users/lenevo/Downloads/erosion.png', erosion)
         cv2.imwrite('C:/Users/lenevo/Downloads/dialation.png', dilation)
         cv2.imwrite('C:/Users/lenevo/Downloads/edges.png', edges)

Out[15]: True
```

5.3.16 Histogram part for edge form of image

```
In [16]: import cv2
         import numpy as np
         from matplotlib import pyplot as plt

         img = cv2.imread('C:/Users/lenevo/Downloads/edges.png',0)

         hist,bins = np.histogram(img.flatten(),256,[0,256])

         cdf = hist.cumsum()
         cdf_normalized = cdf * hist.max()/ cdf.max()

         plt.plot(cdf_normalized, color = 'b')
         plt.hist(img.flatten(),256,[0,256], color = 'r')
         plt.xlim([0,256])
         plt.legend(('cdf','histogram'), loc = 'upper left')
         plt.show()
```



5.3.17 Converting RGB to BGR

Open CV can process objects, faces, and even human handwriting in images and videos. In this article, we will use Python and OpenCV to convert a BGR image to RGB. The BGR image format is used by OpenCV. So, by default, when we read an image with cv2.imread(), it is interpreted in BGR format. The cvtColor() method can be used to convert a BGR image to RGB and vice versa.

```
In [17]: def readb64(base64_string):
             sbuf = BytesIO()
             sbuf.write(base64.b64decode(base64_string))
             pimg = image.open(sbuf)
             return cv2.cvtColor(np.array(pimg), cv2.COLOR_RGB2BGR)
```

5.3.18 Updating the image

```
In [18]: def update_image(pic):
             img = readb64(pic)
             img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
             classification1 = {"Slender":0, "Medium":0, "Bold":0, "Round":0, "Dust":0}
             avg1 = {"Slender":0, "Medium":0, "Bold":0, "Round":0, "Dust":0}
             #convert into binary
             ret,binary = cv2.threshold(img,160,255,cv2.THRESH_BINARY)# 160 - threshold, 255 - value to assign, THRESH_BINARY_INV - Inverse binary
             #averaging filter
             kernel = np.ones((5,5),np.float32)/9
             dst = cv2.filter2D(binary,-1,kernel)# -1 : depth of the destination image

             kernel2 = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3))

             #erosion
             erosion = cv2.erode(dst,kernel2,iterations = 1)

             #dilation
             dilation = cv2.dilate(erosion,kernel2,iterations = 1)

             #edge detection
             edges = cv2.Canny(dilation,100,200)
```

```python
### Size detection
contours, hierarchy = cv2.findContours(erosion, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
#print("No. of rice grains=",len(contours))
total_ar1=0
for cnt in contours:
    x,y,w,h = cv2.boundingRect(cnt)
    aspect_ratio = float(w)/h
    if(aspect_ratio<1):
        aspect_ratio=1/aspect_ratio
    print(round(aspect_ratio,2),get_classification(aspect_ratio))
    classification1[get_classification(aspect_ratio)] += 1
    if get_classification(aspect_ratio) != "Dust":
        total_ar1+=aspect_ratio
    if get_classification(aspect_ratio) != "Dust":
        avg1[get_classification(aspect_ratio)] += aspect_ratio
avg_ar1=total_ar1/len(contours)
if classification1['Slender']!=0:
    avg1['Slender'] = avg1['Slender']/classification1['Slender']
if classification1['Medium']!=0:
    avg1['Medium'] = avg1['Medium']/classification1['Medium']
if classification1['Bold']!=0:
    avg1['Bold'] = avg1['Bold']/classification1['Bold']
if classification1['Round']!=0:
    avg1['Round'] = avg1['Round']/classification1['Round']
cv2.imwrite('C:/Users/lenevo/Downloads/rice2.png', img)
cv2.imwrite('C:/Users/lenevo/Downloads/Binary.png', binary)
cv2.imwrite('C:/Users/lenevo/Downloads/dust.png', dst)
cv2.imwrite('C:/Users/lenevo/Downloads/erosion.png', erosion)
cv2.imwrite('C:/Users/lenevo/Downloads/dialation.png', dilation)
cv2.imwrite('C:/Users/lenevo/Downloads/edges.png', edges)
return classification1,avg1,avg_ar1
```

5.3.19 Average aspect ratio & classification of each grain

In [19]:

```
contours,hierarchy = cv2.findContours(erosion, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
print("Total No of grains is= ")
print(len(contours))
total_ar=0
for cnt in contours:
    x,y,w,h = cv2.boundingRect(cnt)
    aspect_ratio = float(w)/h
    if(aspect_ratio<1):
        aspect_ratio=1/aspect_ratio
    print(round(aspect_ratio,2),get_classification(aspect_ratio))
    total_ar+=aspect_ratio
avg_ar=total_ar/len(contours)
print("Average Aspect Ratio= ",round(avg_ar,2),get_classification(avg_ar))
```

```
Total No of grains is=
52
1.29 Bold
2.0 Bold
1.29 Bold
1.62 Bold
1.78 Bold
2.14 Medium
1.5 Bold
1.0 Round
1.23 Bold
1.25 Bold
1.92 Bold
1.11 Bold
1.73 Bold
1.54 Bold
2.0 Bold
1.25 Bold
1.52 Bold
2.2 Medium
```

```
1.13 Bold
1.0 Round
3.33 Slender
1.91 Bold
2.1 Medium
1.33 Bold
1.62 Bold
1.06 Bold
1.36 Bold
1.08 Bold
3.67 Dust
1.27 Bold
1.2 Bold
1.33 Bold
1.43 Bold
1.23 Bold
1.43 Bold
1.58 Bold
1.36 Bold
1.7 Bold
1.23 Bold
1.18 Bold
2.0 Bold
1.33 Bold
2.33 Medium
2.4 Medium
1.29 Bold
1.22 Bold
1.55 Bold
2.86 Medium
2.1 Medium
1.88 Bold
4.0 Dust
1.2 Bold
Average Aspect Ratio=  1.67 Bold
```

5.3.20 Displaying image

In [20]:

```python
def get_image(path):
    img=image.open(path)
    # Constants
    img_width = 710
    img_height = 550
    scale_factor = 0.5
    fig = go.Figure()
    fig.add_trace(
        go.Scatter(
            x=[0, img_width * scale_factor],
            y=[0, img_height * scale_factor],
            mode="markers",
            marker_opacity=0
        )
    )
    fig.update_xaxes(
        visible=False,
        range=[0, img_width * scale_factor]
    )
    fig.update_yaxes(
        visible=False,
        range=[0, img_height * scale_factor],
        scaleanchor="x"
    )
```

```python
    fig.add_layout_image(
        dict(
            x=0,
            sizex=img_width * scale_factor,
            y=img_height * scale_factor,
            sizey=img_height * scale_factor,
            xref="x",
            yref="y",
            opacity=1.0,
            layer="below",
            sizing="stretch",
            source=img)
    )
    fig.update_layout(
        width=img_width * scale_factor,
        height=img_height * scale_factor,
        margin={"l": 0, "r": 0, "t": 0, "b": 0},
    )
    fig.show(config={'doubleClick': 'reset'})
    return fig
```

5.3.21 Average aspect ratio Vs Classification plot

```
In [21]: def get_plot1(classification = classification, avg = avg, avg_ar = avg_ar):
    fig = subplots.make_subplots(rows=1,cols=1,specs=[[{"type":"bar"}]], shared_xaxes=True)
    print(list(classification.keys()))
    print(list(classification.values()))
    plot1 = go.Bar(x=list(classification.keys()), y=list(classification.values()), name="Particles")
    plot2 = go.Bar(x=list(avg.keys()), y=list(avg.values()), name="Avg. Aspect Ratio")
    fig.add_trace(plot1,1,1)
    fig.add_trace(plot2,1,1)
    fig.add_shape(
        type="line",
        x0=0,
        y0=round(avg_ar,2),
        x1=5,
        y1=round(avg_ar,2),
        line=dict(
            color="LightSeaGreen",
            width=4,
            dash="dashdot",
        ),
    )
    fig.update_layout(
        width = 600,
        height = 350,
        margin = {"l": 5, "r": 5, "t": 30, "b": 5},
        title = "Average Aspect Ratio Vs Classification",
        template = "plotly_dark"
    )
    return fig
```

5.3.22 Quality Analysis

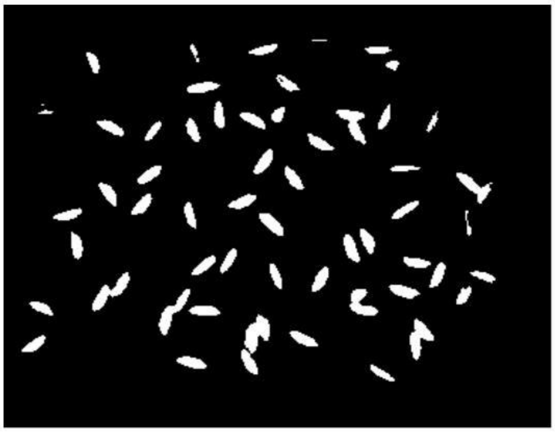
```
In [22]: def get_plot2(classification = classification):
    fig = subplots.make_subplots(rows=1,cols=1,specs=[[{"type":"pie"}]])
    rice = sum(list(classification.values())) - classification['Dust']
    dust = classification['Dust']
    values = [rice, dust]
    labels = ["Rice", "Dust"]
    plot1 = go.Pie(labels=labels, values=values, hole=.3)
    fig.add_trace(plot1,1,1)
    fig.update_layout(
        width = 600,
        height = 350,
        margin = {"l": 65, "r": 5, "t": 60, "b": 50},
        title = "Quality Analysis",
        template = "plotly_dark"
    )
    return fig
```

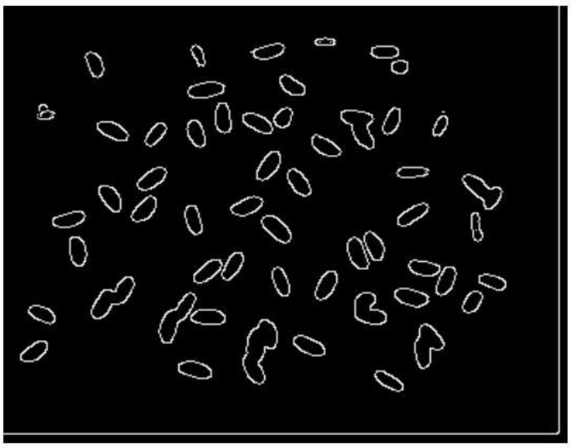
5.3.23 App Layout

```
In [23]: #
app.layout = html.Div([
        html.Div([
                html.Div(style={"float":"left","padding" : "5px 0 5px 50px"}),
                html.Div(
                        children="Classification and Quality Analysis of Rice",
                        style={"float":"left","padding" : "10px 0 10px 10px","font-size": "17px", "font-weight" :"600"}
                ),
        ],className="nav"),
        html.Div([],style={"height":"50px"},id="home"),
        html.Div([
                html.H1(children="Visualisation of Results", style={"text-align":"center", "margin":"0", "padding-bottom" : "20px", "color" : "blue"}),
                html.Div([
                        html.Div([
                                dcc.Graph(figure=get_plot1(),id="graph1"),
                                html.P("Original Image", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"}),
                        html.Div([
                                dcc.Graph(figure=get_plot2(),id="graph2"),
                                html.P("Binary Image", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"}),
                ], style = {"display": "flex", "justify-content": "center", "align-items": "center", "text-align":"center"}),
                html.Div([]),
                html.Div([
                        html.Div([
                                dcc.Upload([
                                        'Drag and Drop or ',
                                        html.A('Select a File')
                                ],
                                style={
                                        'width': '100%',
                                        'height': '60px',
                                        'lineHeight': '60px',
                                        'borderWidth': '1px',
                                        'borderStyle': 'dashed',
                                        'borderRadius': '5px',
                                        'textAlign': 'center'
```

```
                                }, id="upload-image"),
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"}),
                ], style = {"display": "flex", "justify-content": "center", "align-items": "center", "text-align":"center", "width" : "100%"})
        ],style = {"color":"black", "padding" : "20px 0 20px 0", "color" : "black"},id='plots'),
        html.Div([
                html.H1(children="Images", style={"text-align":"center", "margin":"0", "padding-bottom" : "20px"}),
                html.Div([
                        html.Div([
                                dcc.Graph(figure=get_image('C:/Users/lenevo/Downloads/rice2.png'),id="img"),
                                html.P("Original Image", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"}),
                        html.Div([
                                dcc.Graph(figure=get_image('C:/Users/lenevo/Downloads/Binary.png'),id="binary"),
                                html.P("Binary Image", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"}),
                        html.Div([
                                dcc.Graph(figure=get_image('C:/Users/lenevo/Downloads/dust.png'),id="dst"),
                                html.P("Dust Image", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"})
                ], style = {"display": "flex", "justify-content": "center", "align-items": "center", "text-align":"center"}),
                html.Div([]),
                html.Div([
                        html.Div([
                                dcc.Graph(figure=get_image('C:/Users/lenevo/Downloads/erosion.png'),id="erosion"),
                                html.P("Erosion", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"}),
                        html.Div([
                                dcc.Graph(figure=get_image('C:/Users/lenevo/Downloads/dialation.png'),id="dilation"),
                                html.P("Dilation", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"}),
                        html.Div([
                                dcc.Graph(figure=get_image('C:/Users/lenevo/Downloads/edges.png'),id="edges"),
                                html.P("Edge Detection", style={"margin":"0","padding-bottom":"10px"})
                        ], style = {"display": "block", "justify-content": "center", "align-items": "center", "padding":"0 20px 0 20px"})
                ], style = {"display": "flex", "justify-content": "center", "align-items": "center", "text-align":"center"})
        ],style = {"color":"black", "background-color" : "lightsteelblue", "border-radius":"40px 40px 40px 40px", "padding" : "20px 0 20px 0"},id='images')
])
```

```
['Slender', 'Medium', 'Bold', 'Round', 'Dust']
[1, 7, 40, 2, 2]
```



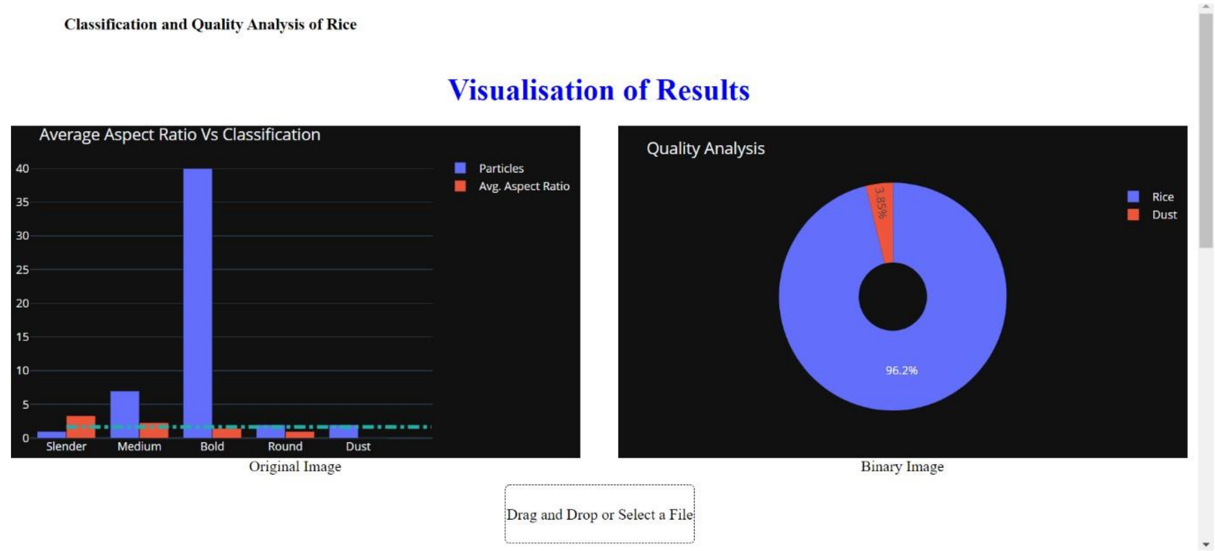


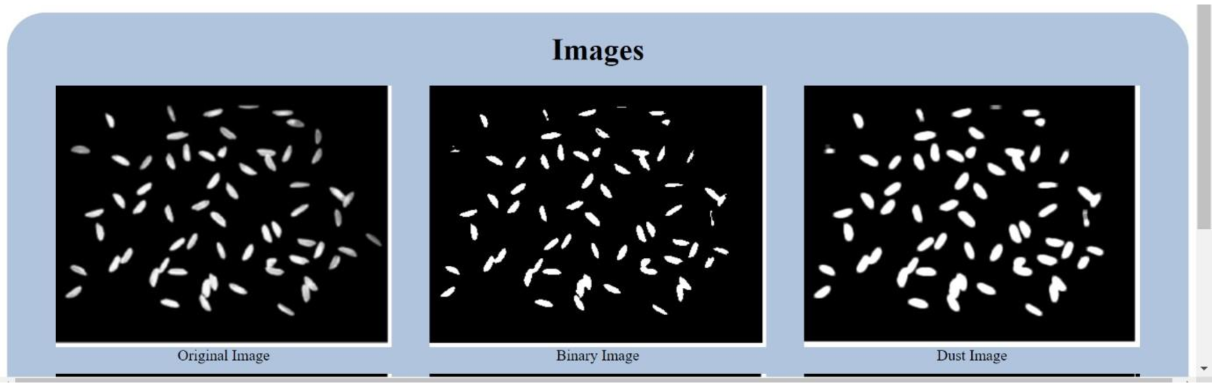
5.3.24 App callback

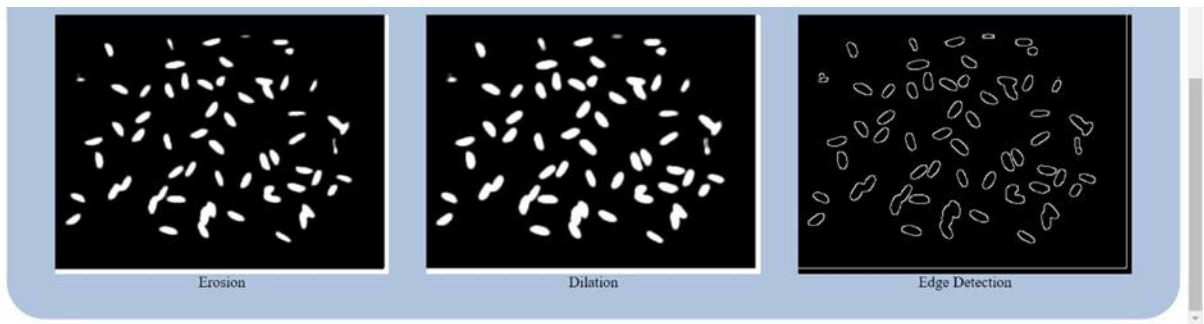
```
In [24]: #app callback
         @app.callback([Output('img', 'figure'),
                                   Output('binary', 'figure'),
                                   Output('dst', 'figure'),
                                   Output('erosion', 'figure'),
                                   Output('dilation', 'figure'),
                                   Output('edges', 'figure'),
                                   Output('graph1', 'figure'),
                                   Output('graph2', 'figure')],
                                  [Input('upload-image', 'contents')]
         )
         #updating the outputs
         def update_output(list_of_contents):
                 if list_of_contents is not None:
                         ind = str(list_of_contents).find(",")
                         cla,av,av_ar = update_image(list_of_contents[ind:])
                         return get_image('C:/Users/lenevo/Downloads/rice2.png'), get_image('C:/Users/lenevo/Downloads/Binary.png'), get_image('C:/Users/lenevo/Downloac
                 else:
                         return get_image('C:/Users/lenevo/Downloads/rice2.png'), get_image('C:/Users/lenevo/Downloads/Binary.png'), get_image('C:/Users/lenevo/Downloac
```

```
In [ ]: if __name__ == '__main__':
                app.run_server(debug=False)

        Dash is running on http://127.0.0.1:8050/

         * Serving Flask app '__main__' (lazy loading)
         * Environment: production
           WARNING: This is a development server. Do not use it in a production deployment.
           Use a production WSGI server instead.
         * Debug mode: off
```

5.4 Results (Accuracy) / Output Screens







6. TESTING

6.1 Types of testing

White-box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Unit Testing:

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules. These tests were carried out during the programming stage itself. All units of Vienna SQL were successfully tested.

Integration testing :

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

System testing:

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and it's original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verification and validations are done by giving input values to the system and by comparing with expected output. Top- down testing implementing here.

Acceptance Testing:

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

6.2 Test cases

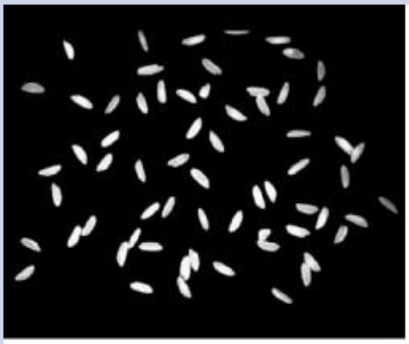
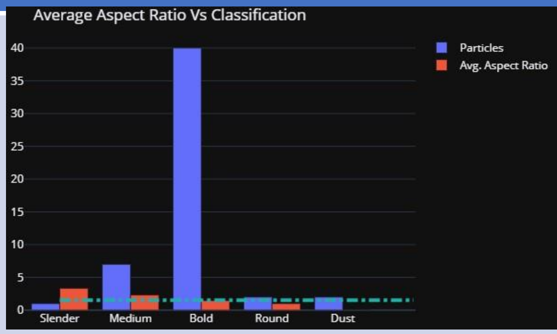
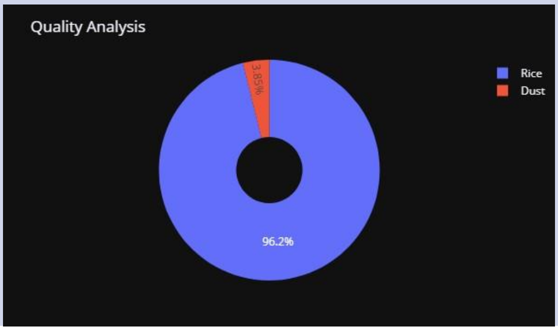
Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

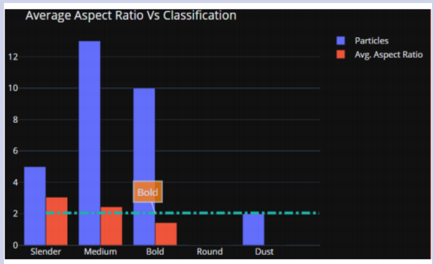
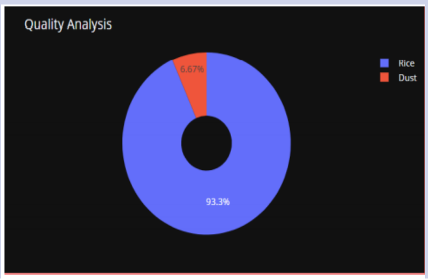
Using White-Box testing methods, the software engineer can drive test cases that

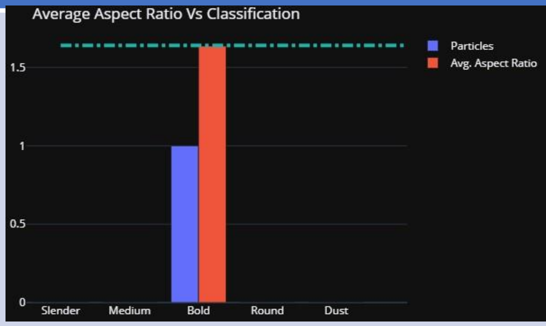
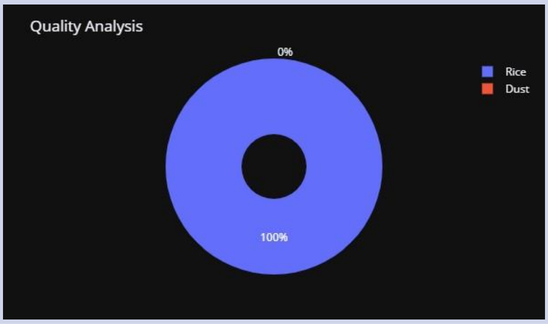
- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and with in their operational bounds.
- Exercise internal data structure to assure their validity.

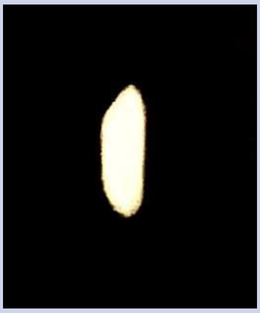
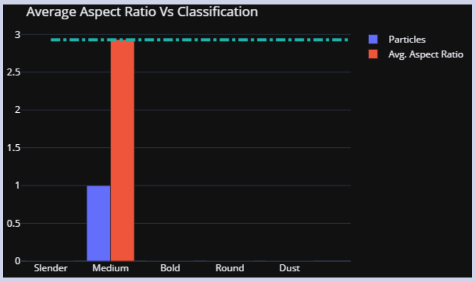
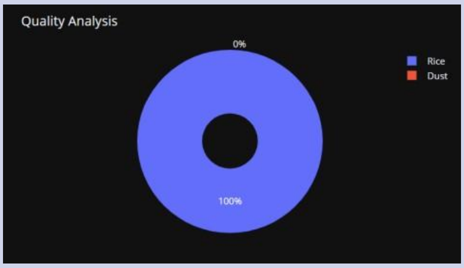
The test case specification for system testing has to be submitted for review before system testing commences.

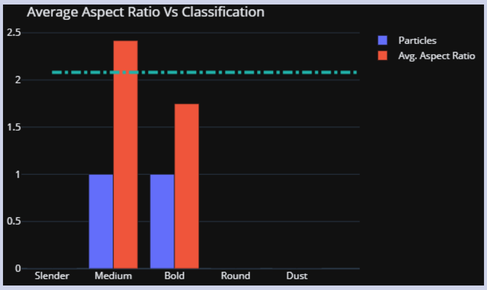
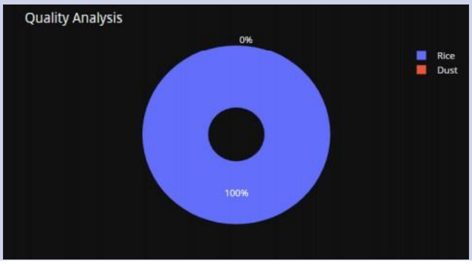
Here in this project , the inputs/test cases are the black background images, we provide an image which is in .jpg or .png format.

S.no	INPUT	OUTPUT
1		 

S.no	INPUT	OUTPUT
2		 

S.no	INPUT	OUTPUT
3		

S.no	INPUT	OUTPUT
4		

S.no	INPUT	OUTPUT
5		 

•Grouped Bar chart – Used for Classification purpose

• Blue Bar indicates the Number of Rice grains.

• Red Bar indicates Average Aspect Ratio.

•Pie chart – Used for Quality Analysis purpose

• Blue Section indicates percentage of Rice grains in the given sample.

• Red Section indicates percentage of Dust in the given sample.

7. CONCLUSION

In this project, we classify the taken rice grain sample into different categories and also analyze its quality based on the aspect ratio, so comparison with other works is not possible. Existing work only detects the rice grains or calculates the number of rice grains in the given sample, but our work helps analyze the quality of the rice sample and place it into a specific category. The quality of the grains in the samples is nearly 100% accurate and capable of efficiently classifying high-value grains, which is otherwise very timeconsuming in manual analysis. This function can save a lot of time and manpower.

8. FUTURE SCOPE

The majority of quality analysis factors must be measured using image processing techniques. This research could be expanded to develop a method for identifying granules based on any attribute that can be used to improve rice quality. The cost of such a system should be low, as should the time spent on quality analysis.

9. BIBILOGRAPHY

[1] "Laboratory Manual on Rice Grain Quality", Directorate of Rice Research, Rajendranagar, Hyderabad, September 2013.

[2] Mahale, Bhagyashree, and Sapana Korde. "Rice quality analysis using image processing techniques." International Conference for Convergence for Technology-2014. IEEE, 2014.

[3] Ali, S.F., Jamil, H., Jamil, R., Torij, I. and Naz, S., 2017, November. "Lowcost solution for rice quality analysis using morphological parameters and its comparison with standard measurements". In 2017 International Multi-topic Conference (INMIC) (pp. 1-6). IEEE.

[4] "Documentation Python". Online: https://www.python.org/doc/

[5] "Documentation Open CV". Online: https://opencv.org/about/

[6] IBPGR-IRRI Rice Advisory Committee, "Descriptors for Rice Oryza Sativa L". International Rice Research Institute and International Board for Plant Genetic Resources, 2020.

[7] Divya Mohan, M.Raj., Semanticscholar 2016, "Quality Analysis of rice grain using ANN and SVM"

APPENDIX-(PYTHON)

About Python

Python is a programming language, and that implies it's a language the two individuals and PCs can comprehend. Python was created by a Dutch programmer named Guido van Rossum, who made the language to tackle a few issues he found in codes of the time.

Python is a deciphered undeniable level programming language for universally useful programming. Made by Guido van Rossumand first delivered in 1991, Python has a plan theory that underscores code meaningfulness, and a linguistic structure that permits developersto communicate ideas in less lines of code, prominently utilizing critical whitespace. It gives builds that empower clear programming on both little and huge scopes.

Python includes a unique sort framework and programmed memory the executives. It upholds numerous programming ideal models, including object-situated, basic, useful and procedural, and has an enormous and exhaustive standard library.

Python mediators are accessible for the majority working frameworks. All c Python, the reference execution of Python, is open source programming and has a local area based improvement model, as do essentially its variation executions. C Python is overseen by the non-benefit

Python translators are accessible for the overwhelming majority working frameworks. All c Python, the reference execution of Python, is open source programming and has a local area based improvement model, as do virtually its variation executions. C Python is overseen by the non-benefit. C Python is overseen by the non-benefit Python Software Foundation.You Can Use Python for Pretty Much Anything

One huge benefit of learning Python is that a universally useful language can be applied in an enormous assortment of ventures. The following are only the absolute most normal fields where Python has tracked down its utilization:Data science

- Scientific and mathematical computing
- Web development
- Computer graphics

- Basic game development
- Mapping and geography (GIS software)

Python Is Widely Used in Data Science

Python's biological system is developing throughout the long term and it's increasingly more equipped for the measurable investigation.

It's the best split the difference among scale and refinement (in wording od information handling).

Python accentuates efficiencyand intelligibility.

Python is utilized by developers that need to dig into informationexamination or apply factual procedures (and by devs that go to information science)

There are a lot of Python logical bundles for information perception, AI, normal language handling, complex information examination and that's only the tip of the iceberg. These elements make Python an extraordinary device for logical registering and a strong option for business bundles like MatLab. The most well known libraries and instruments for information science are:

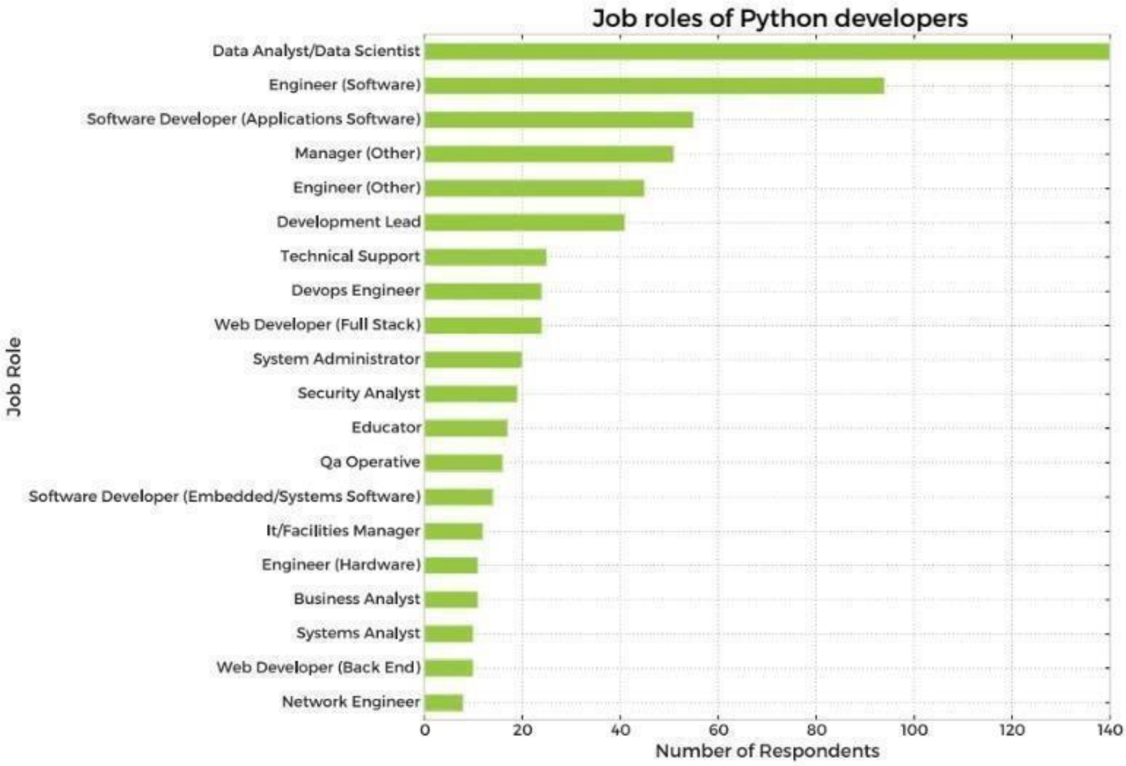
Pandas: a library for information control and investigation. The library gives information designs and tasks to controlling mathematical tables and time series.

NumPy: the major bundle for logical registering with Python, adding support for huge, complex exhibits and frameworks, alongside an enormous library of undeniable level numerical capacities to work on these clusters.

SciPy: a library utilized by researchers, examiners, and specialists doing logical figuring and specialized registering.

Being a free, cross-stage, broadly useful and undeniable level programming language, Python has been generally embraced by mainstream researchers. Researchers esteem Python for its exact and effective punctuation, somewhat level expectation to learn and adapt and the way that it incorporates well with different dialects (for example C/C++).

Because of this ubiquity there are a lot of Python logical bundles for information perception, AI, regular language handling, complex information examination and that's only the tip of the iceberg. These variables make Python an extraordinary device for logical figuring and a strong option for business bundles like Matlab.



Here is our rundown of the most famous Python logical libraries and tools Astropy

The Astropy Project is an assortment of bundles intended for use in cosmology. The center astropy bundle contains usefulness focused on proficient stargazers and astrophysicists, yet might be helpful to anybody creating space science programming.

Biopython

Biopython is an assortment of non-business Python instruments for computational science and bioinformatics. It contains classes to address organic successions and groupingcomments, and it can peruse and keep in touch with an assortment of document designs.

Cubes

Cubes is a light-weight Python system and set of instruments for the improvement of detailing and logical applications, Online Analytical Processing(OLAP), multi-faceted examinationand perusing of totaled information.

DEAP

DEAP is a transformative calculation system for fast prototyping and testing of thoughts. It consolidates the information designs and devices expected to execute most normal developmental calculation methods, for example, hereditary calculation, hereditary programming, advancement techniques, molecule swarm enhancement, differential development and assessment of conveyance calculation.

SCOOP

SCOOP is a Python module for disseminating simultaneous equal errands on different conditions, from heterogeneous frameworks of workstations to supercomputers.

PsychoPy

PsychoPy is a bundle for the age of trials for neuroscience and exploratory brain research. PsychoPy is intended to permit the introduction of boosts and assortment of information for a wide scope of neuroscience, brain research and psychophysics tests.

Pandas

Pandas is a library for information control and examination. The library gives information designs and activities to controlling mathematical tables and time series.

Mlpy

Mlpy is an AI library based on top of NumPy/SciPy, the GNU Scientific Libraries. Mlpy gives a wide scope of AI strategies for regulated and unaided issues and it is pointed toward tracking down a sensible split the difference between seclusion, practicality, reproducibility, convenience and productivity.

Matplotlib

Matplotlib is a python 2D plotting library which produces distribution quality figures in an assortment of printed version designs and intuitive conditions across stages. Matplotlib permits you to produce plots, histograms, power spectra, bar diagrams, errorcharts, scatterplots, and that's only the tip of the iceberg.

NumPy

NumPy is the key bundle for logical processing with Python, adding support for enormous, multi-faceted clusters and networks, alongside a huge library of undeniable level numerical capacities to work on these exhibits.

NetworkX

NetworkX is a library for concentrating on charts which assists you with making, control, and study the construction, elements, and elements of complicated networks.

TomoPy

TomoPy is a publicly released Python tool compartment to perform tomographic information handling and picture recreation assignments. TomoPy gives a cooperative system to the examination of synchrotron tomographic information with the objective to bind together the work of various offices and beamlines performing comparable errands.

Theano

Theano is a mathematical calculation Python library. Theano permits you to characterize, advance, and assess numerical articulations including complex clusters proficiently.

SymPy

SymPy is a library for representative calculation and incorporates highlights going from essential emblematic number-crunching to math, variable based math, discrete science and quantum physical science. It gives PC polynomial math capacities either as an independent application, as a library to different applications, or live on the web.

SciPy

SciPy is a library utilized by researchers, experts, and specialists doing logical processing and specialized registering. SciPy contains modules for streamlining, direct variable based math, mix, addition, extraordinary capacities, FFT, sign and picture handling, ODE solvers and different assignments normal in science and designing.

Scikit-learn

Scikit-learn is an AI library. It highlights different grouping, relapse and bunchingcalculations including support vector machines, arbitrary woodlands, slope helping, k-implies and DBSCAN, and is intended to interoperate with the Python mathematical and logical libraries NumPy and SciPy.

Scikit-image

Scikit-picture is a picture handling library. It incorporates calculations for division, mathematical changes, variety space control, investigation, sifting, morphology, include identification, and that's only the tip of the iceberg.

ScientificPython

ScientificPython is an assortment of modules for logical processing. It contains support for calculation, numerical capacities, measurements, actual units, IO, perception, and parallelization.

SageMath

SageMath is numerical programming with highlights covering numerous parts of science, including variable based math, combinatorics, mathematical arithmetic, number hypothesis, and analytics. SageMath utilizes the Python, supporting procedural, practical and object- arranged develops.

Veusz

Veusz is a logical plotting and charting bundle intended to deliver distribution quality plots in well known vector designs, including PDF, PostScript and SVG.

Graph-tool

Graph-tool is a module for the controland factual examination of charts.

SunPy

SunPy is an information examination climate having some expertise in giving the product important to dissect sun oriented and heliospheric information in Python.

Bokeh

Bokeh is a Python intelligent perception library that objectives current internet browsers for show. Bokeh can help any individual who might want to rapidly and effectively make intuitive plots, dashboards, and information applications. Its will probably give exquisite, brief development of novel illustrations in the style of D3.js, yet in addition convey this ability with elite execution intuitiveness over extremely enormous or streaming datasets.

TensorFlow

TensorFlow is an open source programming library for AI across a scope of undertakings, created by Google to address their issues for frameworks fit for building and preparing brain organizations to distinguish and interpret examples and connections, practically equivalent to the learning and thinking which people use. It is right now utilized for both exploration and creation at Google items, frequently supplantingthe job of its shut source ancestor, DistBelief.

Nilearn

Nilearn is a Python module for quick and simple factual learning on NeuroImaging information. Nilearn makes it simple to utilize many high level AI, design acknowledgment and multivariate factual procedures on neuroimaging information for applications like MVPA (Mutli-Voxel Pattern Analysis), interpreting, prescient demonstrating, practical availability, cerebrum parcellations, connectomes.

Dmelt

DataMelt, or DMelt, is a product for numeric calculation, insights, investigation of huge information volumes ("large information") and logical perception. The program can be utilized in numerous areas, like inherent sciences, designing, displaying and examination of monetary business sectors. DMelt can be utilized with a few prearranging dialects including Python/Jython, BeanShell, Groovy, Ruby, as well likewise with Java.

Python-weka-wrapper

Weka is a set-up of AI programmingwritten in Java, created at the University of Waikato, New Zealand. It contains an assortment of perception devices and calculations for information examination and prescient displaying, along with graphical UIs for simple admittance to these capacities. The python-weka-covering bundle makes it simple to run Weka calculations and channels from inside Python.

Dask

Dask is an adaptable equal figuring library for logical registering made out of two parts: 1) unique undertaking booking upgraded for calculation, enhanced for intelligent computational jobs, and 2) Big Data assortments like equal clusters, dataframes, and records that broaden normal connection points like NumPy, Pandas, or Python iterators to bigger than-memory or dispersed conditions.

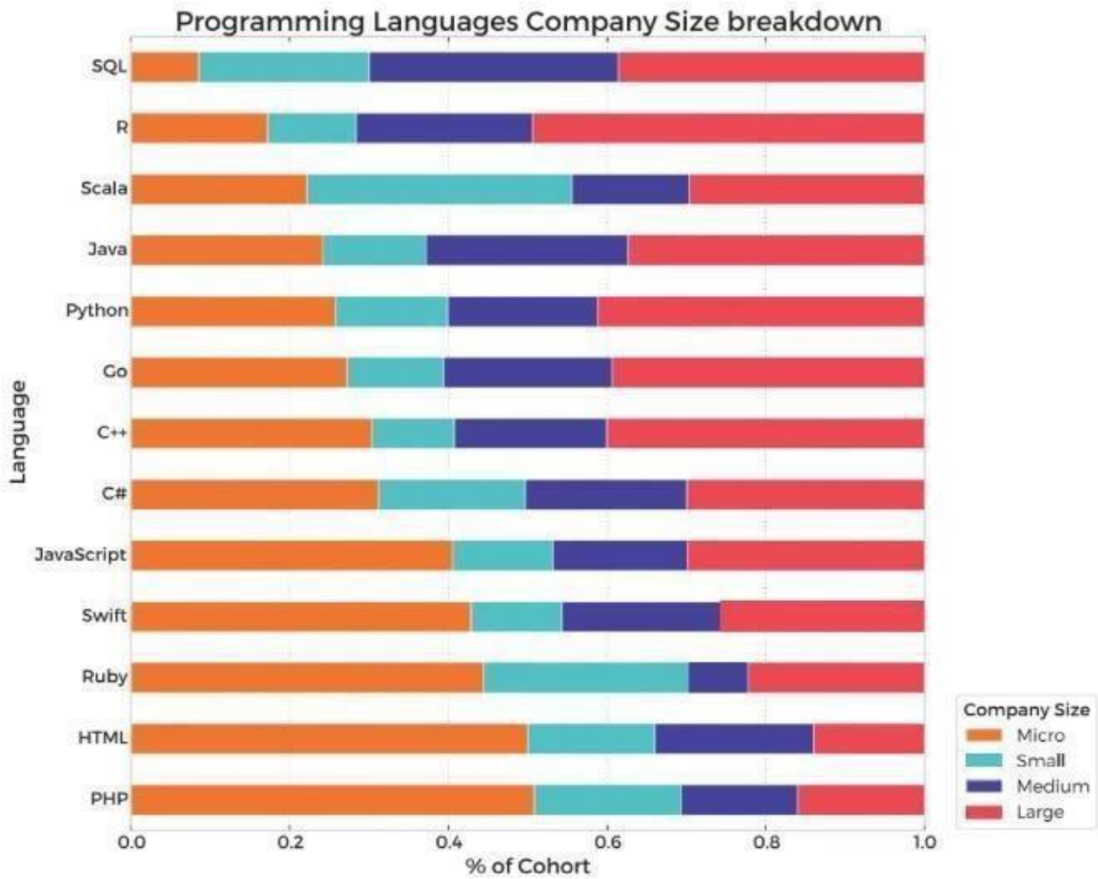
Python Saves Time

Indeed, even the work of art "Hi, world" program outlines this point: print("Hello, world")

For correlation, this is what a similar program resembles in Java: public class HelloWorld {

```
    public static void main(String[] args) { System.out.println("Hello, world");
    }
}
```

All the Big Names Use Python



Python Keywords and Identifier

Keywords are the reserved words in Python.

We cannot use a keyword as variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language.

In Python, keywords are case sensitive.

There are 33 keywords in Python 3.3. This number can varyslightly in course of time.

All the keywords except True, False and None are in lowercase and they must be written as it is. The list of all the keywords is given below.

Keywords in Python programming language

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Identifier is the name given to elements like class, capacities, factors and so on in Python. It helps separating one substance fromanother.

Rules for writing identifiers

Identifiers can be a blend of letters in lowercase (start to finish) or capitalized (a to z) or digits (0 to 9) or a highlight (_). Names like myClass, var_1 and print_this_to_screen, all are substantial model.

An identifier can't begin with a digit. 1variable is invalid, however variable1 is completelyfine.

Keywords cannot be used as identifiers.

```
>>>global = 1
File"<interactive input>", line 1
global = 1
        ^
```

SyntaxError: invalid syntax

We can't utilize extraordinary images like !, @, #, $, % and so on in our identifier.

```
>>> a@ =0
File"<interactive input>", line 1
   a@ =0
^
SyntaxError: invalid syntax
```

Identifier are often of any length.

Python

Python Python includes a very unique sort framework and programmed memory the board. It upholds various programming ideal models, including object-situated, basic, practical and procedural, and has an infinite and extensive standard library.

Python mediators are accessible for the bulk working frameworks. All c Python, the reference execution of Python, is open source programming and incorporates a part area based improvement model, as do essentially its variation executions. C Python is overseen by the non-benefit Python Software Foundation.

Python Logo

DEEP LEARNING:

Deep learning is additionally a machine learning technique that permits computers to hunt out out by example within the identical way that humans do.Deep learning may even be a key component of self-driving cars, allowing them to identify stop signs and distinguish between pedestrians and lampposts. It enables voice control in consumer electronics like phones, tablets, televisions, and hands-free speakers. Deep learning has gotten many press recently, and with good cause. It's accomplishing accomplishments that were previously . In deep learning, a machine model learns to performclassification tasks directly via images, text, or sound.. Deep learning models are able to do state-of-the-art accuracy, even surpassing human performance in some cases. an enormous set of labelled data and neural network topologies are wont to train models.

In a nutshell, accuracy. Deep learning improves recognition accuracy beyond what has ever been achieved previously. this allows consumer electronics to satisfy user expectations, which is important for safety-sensitive applications like self-driving cars. Deep learning has progressed to the aim where it now beats humans in some tasks, like categorising objects in photographs.

While deep learning was first proposed within the 1980s, it's only lately been relevant for 2 reasons:

1. Large volumes of labelled data are required for deep learning. as an example, the creation of self-driving cars involves many photos and thousands of hours of video.
2. Deep learning takes many computational power. The parallel design of high-performance GPUs is good for deep learning. When utilized in conjunctionwith clusters or cloud computing, this unveil a world of possibilities.

Deep learning models are sometimes mentioned as deep neural networks because most deep learning approaches use neural network designs.

Large sets of labelled data and neural network topologies that learn features directly from the knowledge without the requirement for manual feature extraction are accustomed train deep learning models. Convolutional neural networks are one altogether the foremost popular varieties of deep neural networks (CNN or ConvNet).

A CNN uses 2D convolutional layers to mix learnt features with record, making it compatible to processing 2D data like photos. CNNs do away with the requirement for manual feature extraction, so you'll not should determine what features are accustomed classify images.

The CNN operates by extractingfeatures fromphotos directly. The relevant features don't seem to be pre-trained; instead, they're discovered when the network trains on a bunch of images. Deep learning models are particularly accurate for computer vision applications like object categorization because of this automated feature extraction.

Using tens or many hidden layers, CNNs learn to detect different features of a picture. Every hidden layer adds to the complexity of the visual features that are learned. the primary hidden

layer, as an example, could learn to detect edges, while the last word learns to detect more complicated forms that are specific to the form of the thing we're trying to recognise.

PLAGIARISM REPORT

batch17.pdf

ORIGINALITY REPORT

10%	2%	7%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Bhagyashree Mahale, Sapana Korde. "Rice quality analysis using image processing techniques", International Conference for Convergence for Technology-2014, 2014 Publication	6%
2	Submitted to Colorado Technical University Online Student Paper	1%
3	Submitted to UC, Boulder Student Paper	1%
4	www.coursehero.com Internet Source	1%
5	Submitted to Central University of Karnataka Student Paper	<1%
6	pt.scribd.com Internet Source	<1%
7	www.schoolsolver.com Internet Source	<1%
8	docplayer.net Internet Source	<1%

iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 Issue: VI Month of publication: June 2022

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Quality Analysis and Classification of Rice using Image processing

Chityala Udhay[1], Rishabh Semwal[2], Krishna Goel[3], G. Ravi[4]
[1, 2, 3]Students of Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Hyderabad
[4]Associate Professor, Department of Computer Science and Engineering, Sreenidhi Institute of Science andTechnology, Hyderabad

Abstract: Grain quality analysis is a huge challenge in agricultural industries. Internal control is critical in the food industry because food products are characterized and rated into various categories after quality data has been collected. Grain quality assessment is performed by hand, but the results are subjective, lengthy, and pricey. To overcome the limitations and drawbacks of image processing techniques, different resolutions are used for grain quality analysis. Using image processing techniques, this paper proposes a method for grading and analyzing rice based on grain size and form. An edge detection algorithmic software is used in particular to determine the area of each grain's borders. we discover the endpoints of each grain using this technique, and we can then live the grain's endpoints using caliper.
Keywords: Grain quality, rice characteristics, image acquisition, image processing and analysis, grain evaluation, etc.

I. INTRODUCTION

Agricultural enterprise is the oldest and maximum huge enterprise withinside the world. Traditionally, the first-class of meals are described via way of means of a human sensory panel primarily based totally on its bodily and chemical properties. Physical parameters encompass grain length and form, moisture content material, chalk, whiteness, and freeness. For the top-of-line storage, the moisture content material needed to be among 12-14%. Various techniques are used for moisture evaluation.

The primary goal of the proposed method is to provide an alternative approach to quality analysis that requires less time and money. Image processing is an important and advanced technological subject that has seen significant progress. Attempts are being made to replace human manual detection. The document suggests a solution to agribusiness problems.

II. PROBLEM DEFINITION

Product quality analysis is critical in the agriculture sector. An experienced technician visually evaluates the quality of the grain seed. However, the outcome of such an assessment is comparative, varies in results, and takes a long time. The technician's attitude also has an impact on quality; as a result, a new and improved methodology, namely an image processing technique, is presented to address the flaws that have evolved as a result of old ways.

A. Quality and Classification

Grain quality assessment is a significant concern in agriculture. In the food sector, quality control is critical since food is categorized and divided into several classes based on quality factors after harvesting. Grain quality testing is manual, but it is subjective, time-consuming, and expensive. Using image processing techniques, the research provides a method for classifying and grading grains based on grain size and form. Specifically, edge detection to determine each grain's border. We may determine the endpoints of each grain using this technique, and then measure the length and width of the rice using vernier calipers. This process takes very little time and is very affordable.

The image processing technology is used to count the number of rice and classify them based on length, breadth, and length-breadth ratio. The length-width ratio is calculated as follows: length equals the average length of the rice grain, and breadth equals the average breadth of the rice grain.

L/B = [(average rice length) / (average rice breadth)] *100

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

B. Image Acquisition and Processing

A camera is used to capture the image. This is depicted in Figure 1. On the computer, the captured image is saved. Image processing methods are applied to the image after it has been saved.

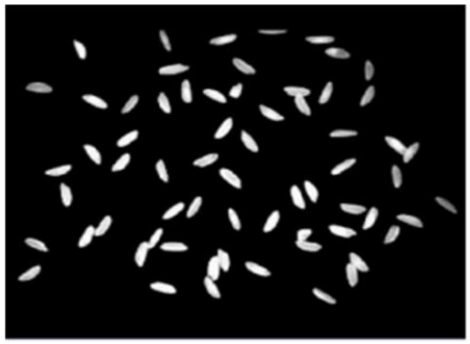


Fig.1 Original Image

III. METHODS

Fig. 2 depicts the flow of the image processing method, which consists of a few basic phases. For image acquisition, rice grains are scattered at random on a black background. The image is saved to be analyzed later. The first phase is pre-processing, which involves image registration and noise removal via a filter. The Shrinkage algorithm is used to segment the touching kernels in the second stage. We have a tendency to use edge detection in the third stage to find the boundary region. The rice grain measurement, as well as length, breadth, and length-breadth measurements, are completed in the fourth stage. Rice is categorized in the fifth stage of the algorithm based on its size and form.

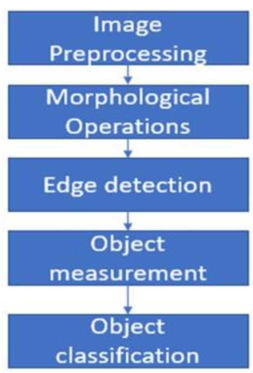


Fig.2 Flow Diagram for Image Processing Algorithm

A. Image Pre-processing

The image acquired with a camera is saved in the computer's 3-D RGB color space, as seen in fig 1. The filter is used to remove noise that happens during the image acquisition process. The image is also sharpened by the filter. The rice grains are segmented from the black background using a threshold technique, and the image is converted to a grey image as seen in fig 3.

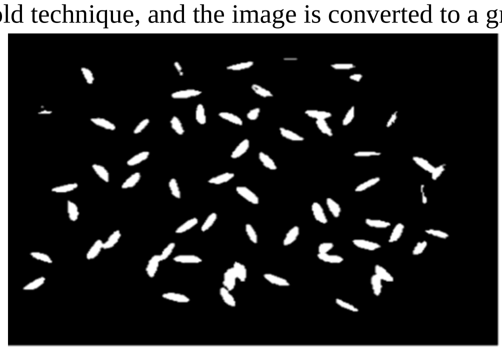


Fig3. Grayscale image

B. Shrinkage Morphological Operation

Rice grains are scattered randomly across a black background. The grains in Figure 1 are not oriented in any way. When contacting grains occur, morphological operations can be used to categorize them. Grain touching can be separated into two types: point and line touching. The combination of dilatation and erosion is a morphological surgery. Erosion is a technique for separating adjacent parts of a grain of rice without compromising its integrity. The erosion process is followed by the dilation process. The purpose of dilatation is to restore the original shape of degraded features without re-joining the divided elements.

In the vision and motion toolbox, there are different types of morphological operations are available such as;

1) Auto M - Auto median,
2) Close - Dilation followed by an erosion,
3) Dilate - Dilation (opposite of erosion),
4) Erode - Erosion which removes isolated background pixels,
5) Open - Erosion followed by dilation,
6) P close - A succession of seven closings and openings,
7) P open - A succession of seven openings and closings.

Fig4. Erosion form of image

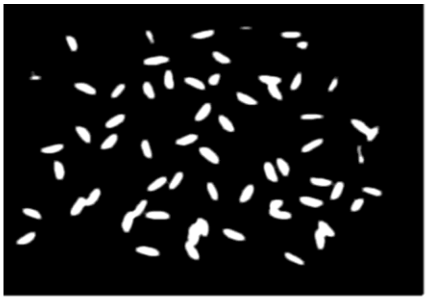


Fig5. Dilation form of image

C. Edge Detection

As illustrated in Fig.6, edge detection aids in locating the region of rice grain boundaries. Gaussian, Gradient, Prewitt, Canny, Fuzzy, and Sobel are six edge detection algorithms offered in Vision and Motion Toolbox. In the proposed methodology, we use the Sobel method to find edges.

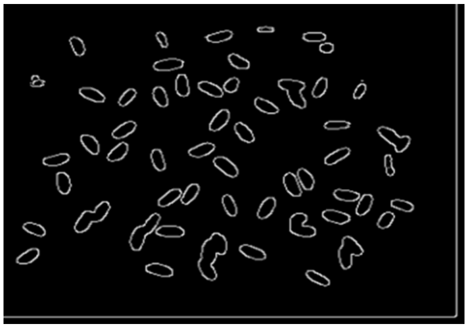


Fig6. Erosion form if the image

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

D. Object Measurement

The number of grains of rice in the image represents the number of individual grains in the evaluation. After counting the number of grains of rice, edge detection techniques are implemented to the image, and endpoint values for each grain are obtained as a result of the algorithm's application. We use a caliper to connect the endpoints and evaluate the length and breadth of every grain. Once we have the length and breadth values, we can calculate the length to breadth ratio.

E. Object Classification

All outcomes must be standard, measured, and calculated to be classified. The laboratory manual on rice grain quality, Board of Rice analysis, Rajendranagar, Hyderabad, provides the standard information for measuring the size and form of rice.

The table below shows how rice grains are classified based on the length and length-to-breadth ratio:

SLENDER	Aspect ratio>=3 and aspect ratio<3.5
MEDIUM	Aspect ratio>=2.1 and aspect ratio<3
BOLD	Aspect ratio>=1.1 and aspect ratio<2.1
ROUND	Aspect ratio>=0.9 and aspect ratio<1
DUST	Aspect ratio>3.5

Table1. Classification Based ON The L/B Ratio

IV. RESULT AND DISCUSSION

Table 4 shows the results obtained from putting image processing algorithms into action. The length-breadth ratio of each grain in the input image is shown in the result.

S.no	Grain Number	L/B ratio	Label
1	Grain 1	1.29	Bold
2	Grain 2	2	Bold
3	Grain 3	1.29	Bold
4	Grain 4	1.62	Bold
5	Grain 5	1.78	Bold
6	Grain 6	2.14	Medium
7	Grain 7	1.5	Bold
8	Grain 8	1	Round
9	Grain 9	1.23	Bold
10	Grain 10	1.25	Bold
11	Grain 11	1.92	Bold
12	Grain 12	1.11	Bold
13	Grain 13	1.73	Bold
14	Grain 14	1.54	Bold
15	Grain 15	2	Bold
16	Grain 16	1.25	Bold
17	Grain 17	1.52	Bold
18	Grain 18	2.2	Medium
19	Grain 19	1.13	Bold
20	Grain 20	1	Round
21	Grain 21	3.33	Slender
22	Grain 22	1.91	Bold

23	Grain 23	2.1	Medium
24	Grain 24	1.33	Bold
25	Grain 25	1.62	Bold
26	Grain 26	1.06	Bold
27	Grain 27	1.36	Bold
28	Grain 28	1.08	Bold
29	Grain 29	3.67	Dust
30	Grain 30	1.27	Bold
31	Grain 31	1.2	Bold
32	Grain 32	1.33	Bold
33	Grain 33	1.43	Bold
34	Grain 34	1.23	Bold
35	Grain 35	1.43	Bold
36	Grain 36	1.58	Bold
37	Grain 37	1.36	Bold
38	Grain 38	1.7	Bold
39	Grain 39	1.23	Bold
40	Grain 40	1.18	Bold
41	Grain 41	2	Bold
42	Grain 42	1.33	Bold
43	Grain 43	2.33	Medium
44	Grain 44	2.4	Medium
45	Grain 45	1.29	Bold
46	Grain 46	1.22	Bold
47	Grain 47	1.55	Bold
48	Grain 48	2.86	Medium
49	Grain 49	2.1	Medium
50	Grain 50	1.88	Bold
51	Grain 51	4	Dust
52	Grain 52	1.2	Bold

Table2. Results for L/B Ratio

Images in which rice grains are randomly arranged and dispersed in a layer are subjected to image analysis techniques. When a fault occurs, such as touching kernels, the shrinking process effectively separates the connecting section from the points where the kernels are touching. Edge detection is used to determine the range of each grain's boundaries and endpoints, after which the length and width can be measured with a caliper. The length-breadth ratio is calculated when the length and breadth values have been determined. A dash app is created to see the results of the Average aspect Ratio Vs Classification chart and a pie chart for Quality analysis of the input image.

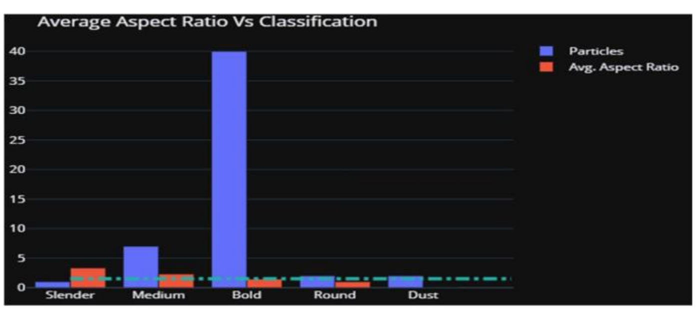


Fig7. Grouped Bar chart - Avg. Aspect Ratio VS Classification

Grouped Bar chart – Used for Classification purposes

- Blue Bar indicates the Number of Rice grains.
- Red Bar indicates Average Aspect Ratio.

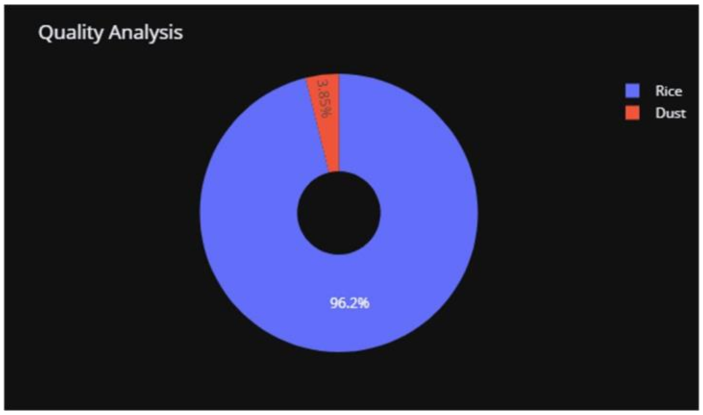


fig8. Pie chart – Quality Analysis

Pie chart – Used for Quality Analysis purposes

- Blue Section indicates the percentage of Rice grains in the given sample.
- Red Section indicates the percentage of Dust in the given sample.

V. CONCLUSION

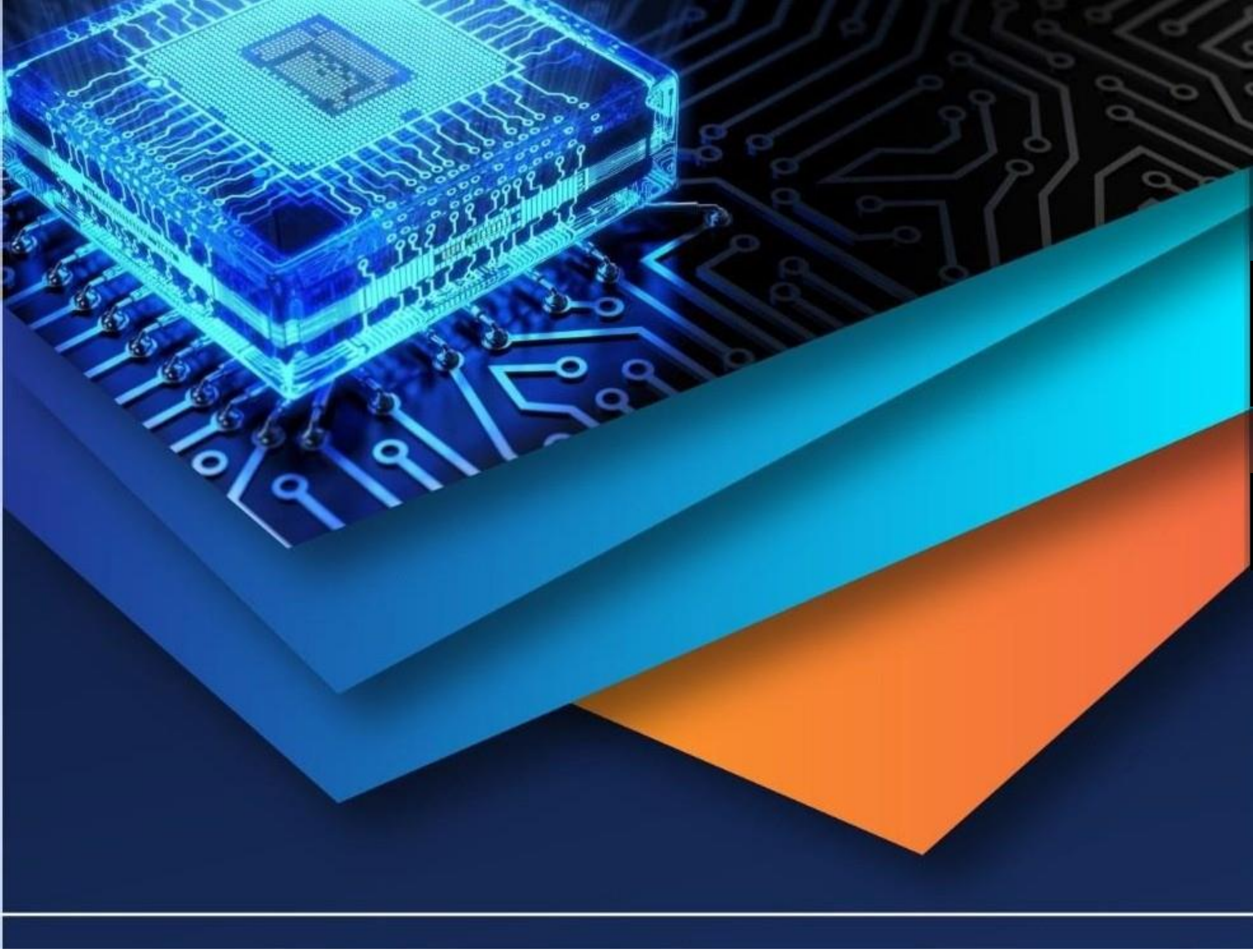
In this project, we classify the taken rice grain sample into different categories and also analyze its quality based on the aspect ratio, so comparison with other works is not possible. Existing work only detects the rice grains or calculates the number of rice grains in the given sample, but our work helps analyze the quality of the rice sample and place it into a specific category. The quality of the grains in the samples is nearly 100% accurate and capable of efficiently classifying high-value grains, which is otherwise very time-consuming in manual analysis. This function can save a lot of time and manpower.

VI. FUTURE WORK

The majority of quality analysis factors must be measured using image processing techniques. This research could be expanded to develop a method for identifying granules based on any attribute that can be used to improve rice quality. The cost of such a system should be low, as should the time spent on quality analysis.

REFERENCES

[1] "Laboratory Manual on Rice Grain Quality", Directorate of Rice Research, Rajendranagar, Hyderabad, September 2013.
[2] Mahale, Bhagyashree, and Sapana Korde. "Rice quality analysis using image processing techniques." International Conference for Convergence for Technology-2014. IEEE, 2014.
[3] Ali, S.F., Jamil, H., Jamil, R., Torij, I. and Naz, S., 2017, November. "Lowcost solution for rice quality analysis using morphological parameters and its comparison with standard measurements". In 2017 International Multi-topic Conference (INMIC) (pp. 1-6). IEEE.
[4] "Documentation Python". Online: https://www.python.org/doc/
[5] "Documentation Open CV". Online: https://opencv.org/about/
[6] IBPGR-IRRI Rice Advisory Committee, "Descriptors for Rice Oryza Sativa L". International Rice Research Institute and International Board for Plant Genetic Resources, 2020.
[7] Divya Mohan, M.Raj., Semanticscholar 2016, "Quality Analysis of rice grain using ANN and SVM".



INDEX COPERNICUS

10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Sreenidhi Institute of Science and Technology Department of Computer Science and Engineering PROJECT-II		
Batch No: E17		**Title**
Roll No	**Name**	
18311A05R4	CH. UDHAY	QUALITY ANALYSIS AND CLASSIFICATION OF RICE USING IMAGE PROCESSING
18311A05W2	RISHABH SEMWAL	
18311A05U3	KRISHNA GOEL	

ABSTRACT

Grain quality analysis is a huge challenge in agricultural industries. Internal control is critical in the food industry because food products are characterized and rated into various categories after quality data has been collected. Grain quality assessment is performed by hand, but the results are subjective, lengthy, and pricey. To overcome the limitations and drawbacks of image processing techniques, different resolutions are used for grain quality analysis. Using image processing techniques, this paper proposes a method for grading and analyzing rice based on grain size and form. An edge detection algorithmic software is used in particular to determine the area of each grain's borders. we discover the endpoints of each grain using this technique.

		Internal Guide	HOD
Student 1	**Ch.Udhay**	**G.Ravi**	**Dr.Aruna Varanasi**
Student 2	**Rishabh Semwal**	**Associate Professor**	**Professor & HOD**
Student 3	**Krishna Goel**	**Department of CSE**	**Department of CSE**

Batch No: E17		Title
Roll No	**Name**	
18311A05R4	CH. UDHAY	QUALITY ANALYSIS AND CLASSIFICATION OF RICE USING IMAGE PROCESSING
18311A05W2	RISHABH SEMWAL	
18311A05U3	KRISHNA GOEL	

Table 1: Project correlation with appropriate POs/PSOs (Please specify level of Correlation, H/M/L against POs/PSOs)

H	High	M	Moderate	L	Low

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY DEPARTMENT OFCOMPUTER SCIENCE AND ENGINEERING Projects Correlation with POs/PSOs														
PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
M	M	L	H	M	L	M	L	M	M	L	M	H	H	H

		Internal Guide	HOD
Student 1	**Ch.Udhay**	**G.Ravi**	**Dr.Aruna Varanasi**
Student 2	**Rishabh Semwal**	**Associate Professor**	**Professor & HOD**
Student 3	**Krishna Goel**	**Department of CSE**	**Department of CSE**

Batch No: E17		**Title**
Roll No	**Name**	
18311A05R4	CH. UDHAY	QUALITY ANALYSIS AND CLASSIFICATION OF RICE USING IMAGE PROCESSING
18311A05W2	RISHABH SEMWAL	
18311A05U3	KRISHNA GOEL	

Table 2: Nature of the Project (Please tick √ Appropriate for your project)

Batch No.	**Title**	**Nature of Project**		
		Product	**Application**	**Research**
E17	QUALITY ANALYSIS AND CLASSIFICATION OF RICE USING IMAGE PROCESSING			✓

		Internal Guide	**HOD**
Student 1	**Ch.Udhay**	**G.Ravi**	**Dr.Aruna Varanasi**
Student 2	**Rishabh Semwal**	**Associate Professor**	**Professor & HOD**
Student 3	**Krishna Goel**	**Department of CSE**	**Department of CSE**

Table 3: Domain of the Project (Please tick √ Appropriate for your project)

Batch No.	**Title**	Domain of the Project				
		ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND DEEP LEARNING	**COMPUTER NETWORKS, INFORMATION SECURITY, CYBER SECURITY**	**DATA WAREHOUSING, DATA MINING, BIG DATA ANALYTICS**	**CLOUD COMPUTING, INTERNET OF THINGS**	**SOFTWARE ENGINEERING, IMAGE PROCESSING**
E17	QUALITY ANALYSIS AND CLASSIFICATION OF RICE USING IMAGE PROCESSING	✓				

		Internal Guide	**HOD**
Student 1	**Ch.Udhay**	**G.Ravi**	**Dr. Aruna Varanasi**
Student 2	**Rishabh Semwal**	**Associate Professor**	**Professor & HOD**
Student 3	**Krishna Goel**	**Department of CSE**	**Department of CSE**