VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi– 590 018

BLDEA's V.P. Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING

AND TECHNOLOGY, VIJAYAPURA

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

A Project Phase-II report on

"RICE GRAIN QUALITY ANALYSIS"

Submitted in partial fulfillment for the award of degree of Bachelor of Engineering in Electronics and Communication Engineering

Submitted by

AMAR HONAWAD	**2BL20EC009**
ANIL HALAGUNAKI	**2BL20EC010**
RAHUL BASETTI	**2BL20EC057**
RAHUL PATIL	**2BL20EC058**

Under the Guidance of

Dr. R M MATH

B.L.D.E.A's

V. P. DR. P. G. HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY, VIJAYAPURA-586103, KARNATAKA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is Certified that the Project Phase-II work entitled "**Rice Grain Quality Analysis**" carried out by **Amar Honawad, Anil Halagunaki, Rahul Basetti, Rahul Patil**, bonafide students of **VP Dr P.G Halakatti College of Engineering and Technology, Vijayapura** in partial fulfillment for the award of **Bachelor of Engineering in Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2023-2024. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project phase-II report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

Internal Guide	HOD	Principal
Dr. R. M. MATH	Dr. U. D. DIXIT	Dr. V. G. SANGAM

External Evaluation

Name of the examiners — Signature with date

1.

2.

B.L.D.E.A's

V. P. DR. P. G. HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY, VIJAYAPURA-586103, KARNATAKA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DECLARATION

We, students of VIII semester B.E, at the department of Electronics & Communication Engineering, hereby declare that, the project phase-II entitled "**Rice Grain Quality Analysis**" embodies the report of our major project work, carried out by us under the guidance of **Dr. R M Math,** We also declare that, to the best of our knowledge and belief, the work reported here in does not form part of any other report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any student.

Place: Vijayapura

Date:

ACKNOWLEDGMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our efforts with success. We consider it as our privilege to express the gratitude to all those who guided in the completion of our Project.

First and foremost, we wish to express our gratitude to our respected Principal **Dr. V G Sangam,** B.L.D. E Association’s V.P. Dr.P.G. Halakatti College of Engineering and Technology, Vijayapura**,** for providing us with a congenial environment to work in.

We would like to express our sincere thanks to **Dr. U D Dixit**, the HOD of Electronics and Communication Engineering, B.L.D. E Association’s V.P. Dr.P.G. Halakatti College of Engineering and Technology, Vijayapura, for his continuous support and encouragement.

We are greatly indebted to our guide **Dr. R M Math**, Department of Electronics and Communication Engineering, B.L.D. E Association’s V.P. Dr.P.G. Halakatti College of Engineering and Technology, Vijayapura, who took great interest in our work. He motivated us and guided us throughout the accomplishment of this goal. We express our profound thanks for his meticulous guidance.

ABSTRACT

Many researches applied machine vision to estimate rice appearance quality inspection. There are various food varieties like rice, wheat, potato, soya bean and maze. Rice is main food crops that all human consumes in all over the world, especially in Asian countries. It is primarily classified according to its grain shape, colour etc. The main crop for our nation to boost agricultural income is grains.

Several contaminants, including stones, weed seeds, chaff, damaged seeds, etc., are present in these grains. Low automation levels and a large human workforce are required for assessing grain quality. Additionally, it increases the cost and length of the testing process.

In order to identify different types of grains and determine the purity of grains using image processing techniques based on various parameters including grain size and shape, we proposed a grain classification system based on machine learning and image processing algorithms.

The Python programming language and Python software are used for all operations machine vision has been used in a most application of grain classification to differentiate rice varieties based on special features such as shape, length, chalkiness, colour and internal damage of rice.

In this paper also discussing and suggesting methods classify four varieties of rice and it also finds the percentage of purity of rice grains using the image processing technics based on several features such as grain colour and shape.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The agricultural industry, spanning across centuries, remains expansive and steeped in tradition. The challenge of assessing grain quality has persisted throughout history. This project introduces a pioneering solution for the evaluation and grading of rice grains by harnessing image processing techniques. Traditionally, the commercial grading of rice hinges on grain size classification, categorizing grains as full, half, or broken. The assessment of food grain quality has conventionally relied on human inspectors employing visual scrutiny. However, the decision-making abilities of human inspectors are susceptible to external influences such as fatigue, subjectivity, and personal biases.

The integration of image processing techniques offers a transformative approach, eliminating the aforementioned challenges while remaining non-destructive and cost-effective. This methodology transcends human limitations, enhancing objectivity and accuracy. The subsequent discussion outlines the procedure deployed to ascertain the percentage quality of rice grains. Rice quality, in essence, is a composite of both physical and chemical attributes. Physical characteristics encompass grain size, shape, chalkiness, whiteness, milling degree, bulk density, and moisture content. On the other hand, chemical attributes involve gelatinization temperature and gel consistency, contributing to the comprehensive assessment of rice quality.

In contrast, conventional methods employed for measuring grain shape and size, such as the grain shape tester, dial micrometre, and graphical method, tend to be protracted and cumbersome. These methods typically allow for the measurement of the dimensions of one grain at a time, yielding results that are not only time-consuming but also susceptible to human errors. Consequently, there is a pressing need for greater precision to fulfill customer expectations and overcome the limitations posed by manual procedures.

On the other hand, the machine vision or the digital image processing is a non-destructive method, it is also very fast and cheap process compared to the chemical method. In the early days of machine vision application to grain quality evaluation, suggested some pattern recognition techniques for identifying and classifying cereal grains.

CHAPTER 2

LITERATURE SURVEY

Food quality detection is a crucial aspect of the food industry, ensuring consumer safety and satisfaction. Recent advancements in machine learning and image processing techniques have revolutionized the accuracy and efficiency of food quality assessment. This literature review aims to provide an in-depth analysis of 15 research papers that explore the integration of machine learning and image processing in food quality detection.

The authors in [1] propose a model to y showcases the application of deep learning techniques, particularly convolutional neural networks (CNNs), for food quality assessment. The authors use image analysis to detect defects, such as mold and discoloration, in food products. The paper [2] focuses on fruit ripeness detection using machine learning algorithms. The authors employ support vector machines (SVM) and random forests to classify fruits into different ripeness categories based on color and texture features. The study [3] presents an automated system for inspecting bakery products' quality. Image processing techniques are combined with support vector machines for real-time detection of defects and anomalies in baked goods.

The research in [4] focuses on fish quality assessment using image analysis and machine learning. The authors in [5] use features like color, texture, and shape to classify fish into different quality categories, ensuring freshness and safety. This paper introduces texture analysis and neural networks for meat quality detection. Texture features extracted from meat images are fed into neural networks to classify meat products based on tenderness and freshness.

This study [6] explores the use of CNNs for detecting diseases and assessing quality in vegetables. The authors develop a model that can identify diseases and quantify the extent of damage using leaf images. Thepaper [7] discusses the application of transfer learning and CNNs for food quality inspection. The authors pre-train a CNN on a large dataset and fine-tune it for specific food quality assessment tasks. The authors in [8] focus on contaminant detection in food products using deep learning techniques. The authors train a CNN to identify foreign objects and contaminants, ensuring food safety.

This study [9] presents a non- invasive approach to inspect egg quality using machine learning. The authors use image analysis and machine learning algorithms to assess egg freshness and defects. The paper [10] introduces an automated system for detecting milk spoilage using image processing and neural networks. The authors [11] employ texture and color features to classify spoiled and fresh milk samples. This research focuses on classifying food items based on image features using decision trees.

The authors extract color, texture, and shape features to develop a decision tree-based classifier. [12] presents a multi-class food quality assessment using deep learning and ensemble methods. The authorsin [13] combine the predictions of multiple models to enhance the accuracy of quality assessment. This paper introduces a hybrid CNN-SVM model for quality inspection of fruits. The authors utilize CNN for feature extraction and SVM for classification, achieving improved accuracy in fruit quality assessment.

This research [14] employs image processing and random forests for dairy product quality detection. The authors use image features to train a random forest model that identifies defects and anomalies in dairy products. This study focuses on automated detection of freshness in seafood using deep learning techniques.The authors use a deep neural network to assess seafood quality based on color, texture, and shape attributes[15].

In conclusion, the reviewed papers collectively highlight the significant advancements achieved in food quality detection through the integration of machine learning and image processing techniques. From deep learning-based approaches to hybrid models, these studies showcase the potential of technology to enhancefood safety, quality, and consumer satisfaction in the food industry.

CHAPTER 3

PROPOSED METHODOLOGY

Utilizing an image processing technique, the assessment of rice seed quantities is undertaken, followed by their classification based on parameters like length, breadth, and the length-breadth ratio. Specifically, the length represents the average longitudinal dimension of rice grains, while breadth pertains to the average width. The length-breadth ratio is computed as L/B = [(Average length of rice grain)/ (average breadth of rice)]*10.

The process is delineated through a series of methodological steps:

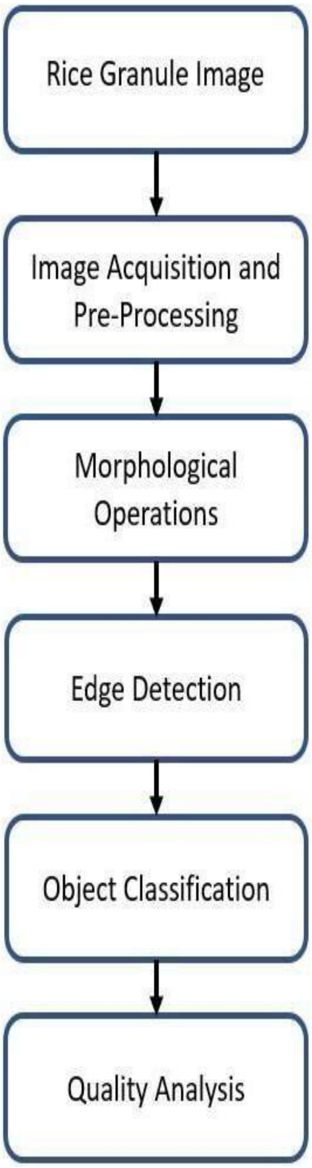

Fig 3.1: Architecture Diagram of the proposed model

a. Image Pre-Processing:

The initial phase involves image pre-processing, during which a filter is applied to eliminate noise generatedduring image acquisition. This filter simultaneously enhances image sharpness. The application of a threshold algorithm aids in segmenting the rice grains from a black background. Filter is applied to removenoise which occurs during the acquisition of image. Filter also sharpens the image. Threshold algorithm is used to segment the rice grains from the black background.

b. Shrinkage Morphological Operation:

Subsequently, a shrinkage morphological operation is employed to address the challenge of segmenting touching rice kernels. The process commences with erosion, which effectively separates interconnected features of rice grains without compromising the integrity of individual ones. Erosion is applied to separatethe touching features of rice grains without losing the integrity of single feature.

c. Edge Detection:

Edge detection, the third step, plays a pivotal role in identifying the boundaries of rice grains. The canny algorithm is adopted for its efficiency in detecting edges. There are six methods are available for edge detection in vision and motion toolbox like differentiation, gradient, permit, Roberts, sigma and Sobel. Themethod specifies the type of edge detection filter to be used. We used Sobel method for edge detection in proposed methodology.

d. Object Measurement:

The fourth stage encompasses object measurement, ascertaining the count of rice grains. Following grain quantification, edge detection algorithms are applied, subsequently yielding endpoint values for each grain.The utilization of a caliper facilitates the connection of endpoints, enabling the measurement of both lengthand breadth. With these dimensions determined, the length-breadth ratio is calculated. We use caliper to jointhe endpoints and measure the value of length and breadth of each grain. After getting the value of length and breadth we can calculate length-breadth ratio.

e. Object Classification:

In the final step of the algorithm, object classification is executed. This necessitates a compilation of standard, measured, and calculated outcomes. In conclusion, the systematic application of image processing techniques, encompassing pre-processing, morphological operations, edge detection, measurement, and classification, forms a comprehensive methodology for accurately quantifying and categorizing rice seeds based on their size and shape attributes.

Grain shape	**L/B ratio**
Slender	Over 3
Medium	2.1 – 3
Bold	1.1 – 2
Round	1 or less

Table 3.1: Classification Criteria based on Aspect Ratio

Aspect Ratio Calculation:

The aspect ratio of each rice grain is calculated using the formula: aspect ratio = width / height.

If the aspect ratio is less than 1, it is normalized to ensure a positive value: aspect ratio = 1 / aspect ratio

Classification Criteria:

The classification of rice grains is based on specific ranges of the aspect ratio. The provided code definesfour categories:

Slender:

If the aspect ratio is greater than or equal to 3, the rice grain is classified as "Slender."

Medium:

If the aspect ratio is between 2.1 and 3 (inclusive), the rice grain is classified as "Medium."

Bold:

If the aspect ratio is between 1.1 and 2.1 (inclusive), the rice grain is classified as "Bold."

Round:

If the aspect ratio is less than or equal to 1, the rice grain is classified as "Round."

However, it represents the distribution of two categories: "Dust Percentage" and "Rice Grain." The chart aims to visually convey the proportion of these two components in the processed image.

Dust Percentage: This represents the percentage of the image covered by dust particles. The code uses image processing techniques to identify and calculate the area occupied by dust. The dust percentage is then displayed in the pie chart.

Rice Grain Percentage: This is the complementary percentage to the dust percentage and represents the area not covered by dust, which is assumed to be occupied by rice grains. It's calculated as 100% minus the dust percentage. Each segment of the pie chart corresponds to either "Dust Percentage" or "Rice Grain Percentage." The chart is labelled accordingly, and the percentage values are displayed on each segment.

In summary, the classification is based on the relative proportions of width and height, and each rice grain is categorized into one of the four classes: Slender, Medium, Bold, or Round. The classification provides insights into the shape characteristics of the rice grains. In this project, we are classifying the rice grain sample taken into various categories and also analysing its quality based on its aspect ratio, so it is not possible to compare with other works.

Output Format:

The classification result is formatted as a string enclosed in parentheses, making it easier to concatenate with other information. In summary, the classification is based on the relative proportions of width and height, and each rice grain is categorized into one of the four classes: Slender, Medium, Bold, or Round. The classification provides insights into the shape characteristics of the rice grains.

CHAPTER 4

SOFTWARE REQUIREMENTS

The Software requirements are as follows:

1) Visual Studio Code:

Visual Studio Code is a lightweight, open-source code editor developed by Microsoft. It provides a modern and customizable interface for software development across multiple platforms.

2) Python:

Python serves as the programming language for the entire application. It provides a clean and readable syntax, making it suitable for rapid development.

3) Flask:

Flask is a micro web framework for Python. It simplifies the process of building web applications by providing tools and libraries for common web development tasks. In this code, Flask is the core framework handling HTTP requests and responses.

4) OpenCV (cv2):

OpenCV, or Open-Source Computer Vision Library, is utilized for computer vision tasks. In this application, OpenCV is used for image processing, including operations like thresholding, filtering, erosion, dilation, and contour detection.

5) NumPy:

NumPy is a powerful library for numerical operations in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions. In this code, NumPy is likely used for efficient array manipulation in the context of image processing.

6) Matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. The code employs Matplotlib to generate a pie chart visualizing dust and rice grain percentages.

7) Web Browser:

The end-users interact with the Flask web application through a web browser. The user interface is generated using HTML templates, and the results are displayed in the browser.

CHAPTER 5

IMPLEMENTATION

5.1) The implementation steps are as follows:

1) Image Processing functions:

a) get classification (ratio):

- This function takes the aspect ratio of a rice grain as input.
- It rounds the aspect ratio to one decimal place.
- It classifies the rice grain based on the rounded aspect ratio into categories: Slender, Medium, Bold, or Round.

b) calculate_dust_percentage(img):

- The classification is appended with parentheses and returned. This function is responsible for calculating the percentage of dust in the input image.
- It applies a threshold to create a binary image, then finds contours to detect dust particles.
- The dust percentage is calculated as the ratio of the dust pixels to the total pixels in the image.

2) Image Processing Pipeline (process image function):

a) Image Filtering and Transformation:

- The function starts by converting the input image to binary using a threshold.
- A 5x5 averaging filter is applied to smooth the binary image.
- Erosion and dilation operations are performed to enhance the features in the image.
- Edge detection using the Canny algorithm is applied.

b) Contour Detection:

- Contours are detected in the processed image to identify individual rice grains.
- The function prints the number of detected rice grains and iterates over each contour to calculate the aspect ratio.

c) Rice Grain Information:

- The aspect ratio, classification, and average aspect ratio of rice grains are printed.
- Information about each rice grain is stored in a list (grain info).

d) Dust Percentage Calculation:

- The calculate_dust_percentage function is called to determine the percentage of dust in the image.•

e) Visualization:

- The processed images are converted to base64 strings for display in the browser.
- A pie chart is generated using Matplotlib to represent the dust percentage and rice grain percentage.

3) Flask Route (@app.route("/")):

a) Handling Form Submissions:

- The route handles both GET and POST requests.
- For GET requests, it renders the index.html template with an image upload form.
- For POST requests, it processes the uploaded image and displays the results on the result.html template.

4) HTML Templates:

a) index.html:

- A simple HTML form for uploading images.

b) result.html:

- Displays processed images, a pie chart, grain information, average aspect ratio, and dust percentage.

5) Running the Application:

- The application is set to run in debug mode for development purposes (app.run(debug=True)).

6) Project Report Integration:

- Highlight the interactive nature of the web application, allowing users to upload images and receive detailed analysis results.
- Emphasize the significance of the classification system and how it aids in categorizing rice grains.
- Discuss the visualizations, including the pie chart, and how they provide a clear representation of dust and rice grain percentages.

CHAPTER 6

ADVANTAGES, APPLICATIONS & LIMITATIONS

6.1) Advantages:

1. **Quality Assurance:** Analysis helps ensure that rice meets predetermined quality standards, ensuring consistency and consumer satisfaction. This is vital for both domestic and international markets.

2. **Nutritional Assessment:** Analysis can determine the nutritional content of rice, including levels of carbohydrates, proteins, fibres, vitamins, and minerals. This information is essential for consumers, especially those with specific dietary requirements.

3. **Marketability:** Understanding the quality of rice helps in positioning it in the market. Premium quality rice can be marketed at higher prices, enhancing profitability for producers and traders.

4. **Consumer Safety:** Analysis can identify contaminants such as heavy metals, pesticides, and mycotoxins, ensuring consumer safety and compliance with food safety regulations.

5. **Processing Optimization:** By analyzing rice quality parameters such as milling degree, chalkiness, and amylose content, processors can optimize their operations to achieve desired outcomes such as better milling yield and improved cooking characteristics.

6. **Varietal Selection:** Quality analysis aids in the selection of rice varieties best suited for specific purposes, such as cooking, milling, or processing into products like rice flour or rice noodles.

7. **Storage and Shelf-life:** Understanding the quality parameters of rice helps in determining appropriate storage conditions and shelf-life, preventing spoilage and quality deterioration over time.

8. **Research and Development:** Quality analysis provides valuable data for research and development purposes, such as breeding programs aimed at developing rice varieties with improved traits and characteristics.

6.2) Applications:

1. **Grain Quality Assessment:** Measures size, shape, and colour for classification based on standards.
2. **Defect Detection:** Identifies broken grains, discolorations, and impurities accurately.
3. **Sorting and Grading:** Automates segregation for different market grades.
4. **Paddy Quality Monitoring:** Evaluates physical attributes of paddy rice before milling.
5. **Quality Control in Processing Plants:** Integrates into processing plants for continuous quality control.
6. **Marketability and Consumer Preference:** Analyzing rice quality allows producers to meet consumer preferences and market demands.
7. **Research and Development:** Grain quality analysis serves as a basis for research on improving rice cultivation practices.
8. **Food Safety and Quality Control:** Analysis ensures that rice meets safety standards and quality requirements for human consumption.

6.3) Limitations:

1. **Subjectivity:** Some quality attributes, such as taste, aroma, and texture, can be subjective and vary based on individual preferences. This subjectivity can lead to inconsistencies in quality assessment, especially when different individuals or organizations are involved in the analysis.

2. **Complexity of Analysis:** Assessing certain quality parameters, such as nutritional content, cooking characteristics, and sensory attributes, may require sophisticated laboratory equipment and trained personnel.

3. **Sampling Variability:** Rice grains within a single batch or lot can exhibit variability in quality due to factors like genetic diversity, environmental conditions, and post-harvest handling practices.

4. **Limited Scope of Analysis:** While various quality parameters can be analyzed individually, evaluating overall grain quality comprehensively can be challenging. Certain important attributes, such as cooking performance and sensory characteristics, may not be adequately captured by standard laboratory tests.

5. **Influence of Processing:** Rice quality can be significantly influenced by post-harvest processing techniques such as milling, polishing, and parboiling.

6. **Influence of Environmental Factors:** The quality of rice grains can be influenced by environmental factors such as soil conditions, climate, and agricultural practices.

7. **Dynamic Nature of Quality:** Rice quality is not static and can change over time due to factors like storage conditions, transportation, and exposure to pests and diseases. Quality analysis conducted at a single point in time may not reflect the dynamic nature of grain quality throughout its lifecycle.

8. **Lack of Standardization:** While there are established standards and protocols for rice quality analysis, these may vary across regions or organizations, leading to inconsistencies in quality assessment and interpretation.

CHAPTER 7

EVALUATION AND RESULT ANALYSIS

The Experimental results are as follows:

The primary goal of this project is to minimize the manual labor involved in the classification of rice grains and the assessment of their quality. To accomplish this, the project leverages the power of Machine Learning and utilizes the Python Flask framework. The culmination of the project results in the creation of a functional website. This web application is capable of receiving input in the form of images depicting rice grains. It then employs Machine Learning techniques to effectively categorize the rice grains and simultaneously evaluate their overall quality.

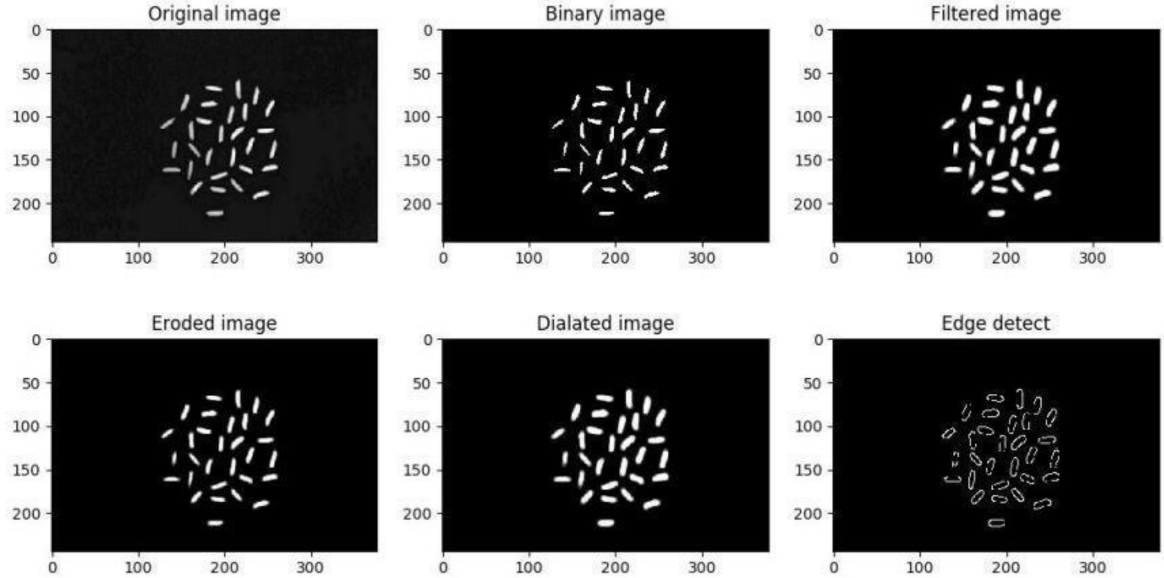

Fig:7.1: Processing of image

Binary Image: A binary image is a type of digital image where each pixel can only have one of two values, usually 0 (black) or 255 (white). It is often used for image segmentation or representing objects in a clear and simple way.

Filtered Image: A filtered image results from applying a filter or convolution operation to the original image. Filters modify the intensity of pixels based on their surrounding values.

Eroded Image: Erosion is a morphological operation that shrinks or thins objects in a binary image. It is achieved by moving a structuring element (a small kernel) over the image and replacing each pixel with the minimum value in its neighbourhood.

Dilated Image: Dilation is a morphological operation that expands or thickens objects in a binary image. It is achieved by moving a structuring element over the image and replacing each pixel with the maximum value in its neighbourhood.

Edge Image: An edge image highlights boundaries or transitions between different regions in an image. It represents areas of rapid intensity change.

However, it represents the distribution of two categories: "Dust Percentage" and "Rice Grain." The chart aims to visually convey the proportion of these two components in the processed image.

Dust Percentage: This represents the percentage of the image covered by dust particles. The code uses image processing techniques to identify and calculate the area occupied by dust. The dust percentage is then displayed in the pie chart.

Rice Grain Percentage: This is the complementary percentage to the dust percentage and represents the area not covered by dust, which is assumed to be occupied by rice grains. It's calculated as 100% minus the dust percentage. Each segment of the pie chart corresponds to either "Dust Percentage" or "Rice Grain Percentage." The chart is labelled accordingly, and the percentage values are displayed on each segment.
The colours used in the pie chart are "light coral" for dust and "lightskyblue " for rice grains.

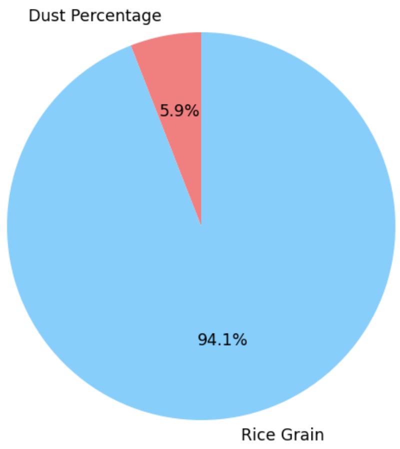

Fig 7.2: Pie chart

A grouped bar chart is employed in this context to facilitate the classification process. The chart effectively presents information related to the classification of rice grains. Notably, the chart employs two distinct bars: The blue bar is indicative of the total count of rice grains within the dataset. The red bar, on the other hand, conveys the average aspect ratio of the rice grains.

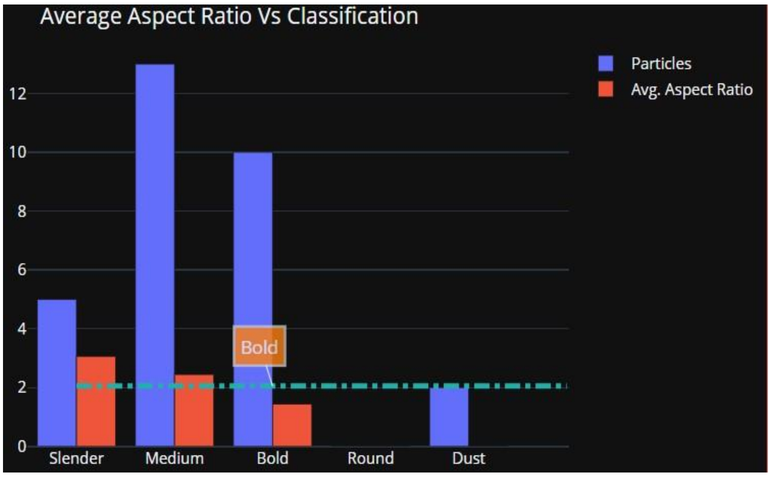

Fig 7.3: Aspect ratio vs Classification

Both the grouped bar chart and the pie chart play integral roles in conveying vital information regarding the classification and quality assessment of rice grains, respectively. Through visual representation, these graphical elements enhance the comprehensibility and insightfulness of the data analysis process.

The classification of rice grains is based on specific ranges of the aspect ratio. The provided code defines four categories:

a) **Slender:** If the aspect ratio is greater than or equal to 3, the rice grain is classified as "Slender."
b) **Medium:** If the aspect ratio is between 2.1 and 3 (inclusive), the rice grain is classified as “Medium.”
c) **Bold:** If the aspect ratio is between 1.1 and 2.1 (inclusive), the rice grain is classified as "Bold."
d) **Round:** If the aspect ratio is less than or equal to 1, the rice grain is classified as "Round.

In summary, the classification is based on the relative proportions of width and height, and each rice grain is categorized into one of the four classes: Slender, Medium, Bold, or Round. The classification provides insights into the shape characteristics of the rice grains. In this project, we are classifying the rice grain sample taken into various categories and also analysing its quality based on its aspect ratio, so it is not possible to compare with other works.

Existing works only detect the rice grains, or calculate number of rice grains in the given sample but our work helps to analyse the quality of rice sample and classify them into particular category.

Aspect Ratio	Type
1.44	(Bold)
2.21	(Medium)
4.07	(Slender)
2.42	(Medium)
1.17	(Bold)
3.56	(Slender)
1.38	(Bold)
1.5	(Bold)
1.91	(Bold)
1.45	(Bold)
1.76	(Bold)
2.3	(Medium)
1.47	(Bold)
1.0	(Round)
4.75	(Slender)
1.06	(Bold)
1.13	(Bold)
1.51	(Bold)

Fig 7.4: Classification of rice in different categories

Grain Percentage

Dust Percentage: 5.9105784351226145%

Rice Grain Percentage: 94.08942156487738%

Fig 7.5: Calculation of Grain Percentage

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Conclusion:

In this project, our focus lies in the comprehensive classification of rice grain samples, coupled with a meticulous analysis of their quality based on the aspect ratio. Our approach distinctly differentiates itself from existing works, as it not only identifies rice grains and quantifies their numbers but also delves deeper to evaluate their quality and allocate them to specific categories This is of paramount importance, particularly for scenarios involving the efficient grading of a large volume of grains.

Our methodology significantly expedites this process, alleviating the substantial time and human effort typically associated with manual analysis. Our image analysis algorithms are applied to images featuring rice grains arranged randomly in a single layer.

Edge detection is subsequently employed to pinpoint boundary regions and determine the endpoints of each individual grain. Subsequently, using a caliper, we measure the length and breadth of each grain. These measurements further allow for the calculation of the length-breadth ratio.

The deployment of these algorithms proves highly efficient in evaluating grain quality based on their size. The paramount advantage of our proposed method is its expedited process, minimal time requirement, cost-effectiveness, and superior performance compared to traditional manual methods.

Future Scope:

For quality analysis, maximum numbers of parameters are to be measured by image processing techniques. Expansion on this work can target to design such a system which can classify rice grains on the basis of each parameter which can used to enhance the quality of rice. The cost of such system should be less and minimize time requirement for quality analysis.

Overall, the future of rice grain quality analysis lies in harnessing interdisciplinary approaches, leveraging cutting-edge technologies, and integrating stakeholder perspectives to ensure safe, nutritious, and high-quality rice production that meets the diverse needs of consumers and contributes to global food security.

REFERENCES

[1] M. B. Lam, T. -H. Nguyen and W. -Y. Chung, "Deep Learning-Based Food Quality Estimation Using Radio Frequency-Powered Sensor Mote," in IEEE Access, vol. 8, pp. 88360-88371, 2020,

[2] S. Gayathri, T. U. Ujwala, C. V. Vinusha, N. R. Pauline and D. B. Tharunika, "Detection of Papaya Ripeness Using Deep Learning Approach," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp.

[3] Biglari, Roya & Mohanna, Farahnaz & Ahsani, Mohammad. (2022). Introducing an Automatic Bread Quality Assessment Algorithm using Image Processing Techniques. European Journal of Electrical Engineering and Computer Science. 6. 31-38.

[4] D. R. Wijaya, N. F. Syarwan, M. A. Nugraha, D. Ananda, T. Fahrudin and R. Handayani, "Seafood Quality Detection Using Electronic Nose and Machine Learning Algorithms With Hyperparameter Optimization," in IEEE Access, vol. 11, pp. 62484-62495, 2023

[5] Shiranita, K. & Hayashi, K. & Otsubo, A. & Miyajima, T. & Takiyama, R. (2000). Determination of meat quality by image processing and neural network techniques. 2.

[6] P. Khobragade, A. Shriwas, S. Shinde, A. Mane and A. Padole, "Potato Leaf Disease Detection Using CNN," 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2022, pp.

[7] S. S. Alahmari and T. Salem, "Food State Recognition Using Deep Learning," in IEEE Access, vol. 10, pp.

[8] L. Urbinati, M. Ricci, G. Turvani, J. A. T. Vasquez, F. Vipiana and M. R. Casu, "A Machine-Learning Based Microwave Sensing Approach to Food Contaminant Detection," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 2020, pp.

[9] L. Geng, H. Wang, Z. Xiao, F. Zhang, J. Wu and Y. Liu, "Fully Convolutional Network with Gated Recurrent Unit for Hatching Egg Activity Classification," in IEEE Access, vol. 7, pp.

[10] S. Karthika Shree, V. Vijayarajan, B. Sathya Bama and S. Mohammed Mansoor Roomi, "Milk Quality Inspection Using Hyperspectral Imaging," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp.

[11] M. Chun, H. Jeong, H. Lee, T. Yoo and H. Jung, "Development of Korean Food Image Classification Model Using Public Food Image Dataset and Deep Learning Methods," in IEEE Access, vol. 10, pp.

[12] F. S. Konstantakopoulos, E. I. Georga and D. I. Fotiadis, "A Review of Image-based Food Recognition and Volume Estimation Artificial Intelligence Systems," in IEEE Reviews in Biomedical Engineering, doi: 10.1109/RBME.2023.3283149.

[13] M. Haggag, S. Abdelhay, A. Mecheter, S. Gowid, F. Musharavati and S. Ghani, "An Intelligent Hybrid Experimental-Based Deep Learning Algorithm for Tomato-Sorting Controllers," in IEEE Access, vol. 7, pp.

[14] Alvarado, Witman & Meneses-Claudio, Brian & RomanGonzalez, Avid. (2019). Milk Purity Recognition Software through Image Processing. International Journal of Advanced Computer Science and Applications.

[15] K. Prema and J. Visumathi, "An Improved NonDestructive Shrimp Freshness Detection Method Based on Hybrid CNN and SVM with GAN Augmentation," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2022, pp.

APPENDEX

Codes and Standards:

Following is the main code for application of rice grain quality analysis:

```
from flask import Flask, render_template, request
import cv2
import numpy as np
import base64
import matplotlib.pyplot as plt
app = Flask(__name__)
def get classification(ratio):
    ratio = round(ratio, 1)
    to_ret = ""
    if ratio >= 3:
        to_ret = "Slender"
    elif 2.1 <= ratio < 3:
        to_ret = "Medium"
    elif 1.1 <= ratio < 2.1:
        to_ret = "Bold"
    elif ratio <= 1:
        to_ret = "Round"
    to_ret = "(" + to_ret + ")"
    return to_ret
def calculate_dust_percentage(img):
    # Your dust detection logic goes here
    # For example, you can apply thresholding and contour detection to identify dust particles
    _, binary_dust = cv2.threshold(img, 200, 255, cv2.THRESH_BINARY)
# Adjust the threshold as needed
# Find contours for dust detection
    contours_dust, _ = cv2.findContours(binary_dust, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    # Calculate dust percentage
    total_pixels = img.size
    dust_pixels = sum(cv2.contourArea(cnt) for cnt in contours_dust)
```

```
    dust_percentage = (dust_pixels / total_pixels) * 100
    return dust_percentage
def process_image(img):
    # Convert the processed image to binary
    _, binary = cv2.threshold(img, 160, 255, cv2.THRESH_BINARY)
    # Apply an 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))
    # Perform erosion
    erosion = cv2.erode(dst, kernel2, iterations=1)
    # Perform dilation
    dilation = cv2.dilate(erosion, kernel2, iterations=1)
    # Apply edge detection
    edges = cv2.Canny(dilation, 100, 200)
    # Find contours for size detection
    contours, _ = cv2.findContours(erosion, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    print("No. of rice grains=", 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))
    # Calculate dust percentage
    dust_percentage = calculate_dust_percentage(img)
    print("Dust Percentage =", round(dust_percentage, 2), "%")
    # Calculate rice grain percentage
    rice_grain_percentage = 100 - dust_percentage
```

```
# Convert the processed image to base64 for displaying in the browser
_, buffer = cv2.imencode('.jpg', img)
img_str = base64.b64encode(buffer).decode('utf-8')
_, buffer1 = cv2.imencode('.jpg', binary)
filter_img_str1 = base64.b64encode(buffer1).decode('utf-8')
_, buffer2 = cv2.imencode('.jpg', dst)
filter_img_str2 = base64.b64encode(buffer2).decode('utf-8')
_, buffer3 = cv2.imencode('.jpg', erosion)
filter_img_str3 = base64.b64encode(buffer3).decode('utf-8')
_, buffer4 = cv2.imencode('.jpg', dilation)
filter_img_str4 = base64.b64encode(buffer4).decode('utf-8')
_, buffer5 = cv2.imencode('.jpg', edges)
filter_img_str5 = base64.b64encode(buffer5).decode('utf-8')
# Collect information about rice grains and aspect ratios
grain_info = []
grain_counts = {}  # Dictionary to store counts for each grain type
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
    grain_type = get_classification(aspect_ratio)
    grain_info.append((round(aspect_ratio, 2), grain_type))
    # Increment the count for the grain type
    if grain_type not in grain_counts:
        grain_counts[grain_type] = 1
    else:
        grain_counts[grain_type] += 1
# Calculate average aspect ratio
avg_aspect_ratio = round(avg_ar, 2)
# Data
labels = ['Dust Percentage', 'Rice Grain']
sizes = [dust_percentage, rice_grain_percentage]
colors = ['lightcoral', 'lightskyblue']
```

```
# Plotting the Pie chart
    fig, ax = plt.subplots()
    ax.pie(sizes, labels=labels, autopct='%1.1f%%', colors=colors, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle.
    ax.axis('equal')
# Save the pie chart as an image
    plt.savefig('pie_chart.png')
# Display the pie chart
    plt.show()
    # Return the base64 strings, grain counts, and dust percentage
    return img_str, filter_img_str1, filter_img_str2, filter_img_str3, filter_img_str4,
filter_img_str5, grain_info, avg_aspect_ratio, grain_counts, dust_percentage,
rice_grain_percentage
@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":
        # Get the uploaded image
        image = request.files["image"]
        if image:
            # Read the image and convert it to grayscale
            img = cv2.imdecode(np.fromstring(image.read(), np.uint8),
cv2.IMREAD_GRAYSCALE)
            # Process the image and get the base64 string, grain counts, and dust percentage
            processed_image = process_image(img)
            return render_template("result.html", processed_image=processed_image,
get_classification=get_classification)
    return render_template("index.html")
if __name__ == "__main__":
    app.run(debug=True)
```

Following is the code for generating results in the application:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <title>Image Processing Result</title>
</head>
<body style="font-family: 'Comic Sans MS', cursive; background: linear-gradient(135deg, #FFB6C1, #87CEFA); color: #333; text-align: center; padding: 20px;">

    <h1 style="color: #3498db; font-size: 2em;">Processed Image</h1>
    <img src="data:image/jpeg;base64,{{ processed_image[0] }}" alt="Processed Image" style="max-width: 100%; height: auto; box-shadow: 0 0 15px rgba(0, 0, 0, 0.2); border-radius: 15px; margin-bottom: 20px;">

    <h2 style="color: #e74c3c; margin-top: 20px; font-size: 1.8em;">Filter Images</h2>
    <div style="display: flex; justify-content: space-around; align-items: center; flex-wrap: wrap; margin-top: 15px;">
        <img src="data:image/jpeg;base64,{{ processed_image[1] }}" alt="Filtered Image 1" style="max-width: 18%; height: auto; border-radius: 15px; margin-bottom: 15px;">
        <img src="data:image/jpeg;base64,{{ processed_image[2] }}" alt="Filtered Image 2" style="max-width: 18%; height: auto; border-radius: 15px; margin-bottom: 15px;">
        <img src="data:image/jpeg;base64,{{ processed_image[3] }}" alt="Filtered Image 3" style="max-width: 18%; height: auto; border-radius: 15px; margin-bottom: 15px;">
        <img src="data:image/jpeg;base64,{{ processed_image[4] }}" alt="Filtered Image 4" style="max-width: 18%; height: auto; border-radius: 15px; margin-bottom: 15px;">
        <img src="data:image/jpeg;base64,{{ processed_image[5] }}" alt="Filtered Image 5" style="max-width: 18%; height: auto; border-radius: 15px; margin-bottom: 15px;">
        <!-- Add more as needed -->
    </div>
```

```
    {% if processed_image[6] %}
        <h2 style="color: #27ae60; margin-top: 20px; font-size: 1.8em;">Rice Grain
Information</h2>
        <table border="1" style="margin: 20px auto; border-collapse: collapse; background-color:
#fff; border-radius: 15px; box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);">
            <thead>
                <tr>
                    <th style="padding: 15px; background-color: #3498db; color: #fff;">Aspect
Ratio</th>
                    <th style="padding: 15px; background-color: #3498db; color: #fff;">Type</th>
                </tr>
            </thead>
            <tbody>
                {% for info in processed_image[6] %}
                    <tr>
                        <td style="padding: 15px;">{{ info[0] }}</td>
                        <td style="padding: 15px;">{{ info[1] }}</td>
                    </tr>
                {% endfor %}
            </tbody>
        </table>
        <p style="font-size: 2em; color: #333; margin-bottom: 30px; font-weight: bold;">Average
Aspect Ratio: {{ processed_image[7] }} ({{ get_classification(processed_image[7]) }})</p>
    {% endif %}
<!-- Add this inside the body where you want the chart to appear -->
<canvas id="riceChart" style="width: 20%;"></canvas>
<!-- Add this script section after including Chart.js -->
<script>
    // Get the data from the server-side
    var counts = JSON.parse('{{ processed_image[8]|tojson }}');
    // Create a bar chart
    var ctx = document.getElementById('riceChart').getContext('2d');
    var myChart = new Chart(ctx, {
        type: 'bar',
```

```
        data: {
            labels: Object.keys(counts),
            datasets: [{
                label: 'Number of Grains',
                data: Object.values(counts),
                backgroundColor: 'rgba(75, 192, 192, 0.2)',
                borderColor: 'rgba(75, 192, 192, 1)',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true
                }
            }
        }
    });
</script>
{% if processed_image %}
    <!-- ... Other content ... -->
    <h2 style="color: #e74c3c; margin-top: 20px; font-size: 1.8em;">Grain Percentage</h2>
    <p style="font-size: 2em; color: #333; margin-bottom: 30px; font-weight: bold;">Dust
Percentage: {{ processed_image[9] }}%</p>
    <p style="font-size: 2em; color: #333; margin-bottom: 30px; font-weight: bold;">Rice Grain
Percentage: {{ processed_image[10] }}%</p>
{% endif %}
</body>
</html>
```