



**RED KIDNEY BEAN (RKB) CLASSIFICATION AND GRADING
USING IMAGE PROCESSING TECHNIQUES**

A Thesis Presented

by

Yared Getnet

to

The Faculty of Informatics

of

St. Mary's University

**In Partial Fulfillment of the Requirements
for the Degree of Master of Science**

in

Computer Science

February 2022

ACCEPTANCE

**Red Kidney Bean (RKB) Classification and Grading
Using Image Processing Techniques**

By

Yared Getnet

**Accepted by the Faculty of Informatics, St. Mary's University, in partial
fulfillment of the requirements for the degree of Master of Science in
Computer Science**

Thesis Examination Committee:

Internal Examiner

External Examiner

Dean, Faculty of Informatics

February 2022

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Yared Getnet
Full Name of Student

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Million Meshesha
Full Name of Advisor



Signature

Addis Ababa

Ethiopia

February 02, 2022

Acknowledgement

I would like to express my deepest gratitude to my advisor, Dr. Million Meshesha for his excellent guidance, caring, and patience during the thesis.

I am also grateful to the Ethiopian Commodity Exchange, Ato Aweke & Ato Yenebebe of ECX Warehouse Laboratory, and Ato Belay of Ethiopian Commodity Exchange Authority for their insightful comments and support.

Table of Contents

List of Acronyms.....	viii
List of Figures	ix
List of Tables	x
Abstract	xi
CHAPTER ONE.....	1
Introduction	1
1.1 Background	1
1.2 Motivation.....	2
1.3 Statement of the Problem.....	2
1.4 Objective of the Study	4
1.4.1 General Objective	4
1.4.2 Specific Objective.....	4
1.5 Scope and Limitation of the Study	4
1.6 Significance of the Research	4
1.7 Methodology	5
1.7.1 Research Design	5
1.7.2 Data Collection & Preparation.....	5
1.7.3 Implementation Tools.....	6
1.7.4 Evaluation Method	6
1.8 Organization of the Thesis	7
CHAPTER TWO.....	8
Literature Review	8
2.1 Overview	8
2.2 Red kidney Bean Sample Constituents	8
2.3 Digital Image Representation	9
2.4 Digital Image Processing	11
2.4.1 Image Preprocessing.....	11
2.4.2 Image Segmentation	12
2.4.2.1 Histogram Based Thresholding	12

2.4.2.2 Edge-Based Segmentation	13
2.4.3 Feature Extraction	14
2.4.4 Classification	15
2.5.3 Research Gap	23
CHAPTER THREE	24
Methods	24
3.1 Overview	24
3.2 The Proposed Architecture	24
3.3 RKB Image Data Collection	26
3.4 Image Pre-Processing	26
3.5 Image Segmentation	27
3.5.1 Segmentation Using Thresholding	28
3.5.2. Merging Segmented Images	30
3.6 Feature Extraction	31
3.6.1 Color Feature Extraction	31
3.6.2 Size Feature Extraction	34
3.6.3 Shape Feature Extraction	35
3.7 Classification	36
3.7.1 Naïve Bayes Classifier	38
3.8 Grading of RKB	39
3.9 Evaluation Methods	40
3.10 Summary	41
CHAPTER FOUR	42
Experimentation and Discussion	42
4.1 Overview	42
4.2 Data Set	42
4.3 Experimenting Segmentation	43
4.4 Test Results of Classification	46
4.4.1 Experimental Setup	46
4.4.2 Naïve Bayes Classifier Test Results	47

4.4.3 ANN Classifier Test Results	48
4.5 Discussion of Result	52
5.2 Contribution to Knowledge	55
CHAPTER FIVE	56
Conclusion and Future Work	56
5.1 Conclusion	56
5.2 Future Work	57
References	58
Appendices A - System Code	63
Appendices B - Sample Red Kidney Bean (RKB) Images used for Classification and Grading	74

List of Acronyms

ANN	Artificial Neural Network
ECX	Ethiopian Commodity Exchange
FD	Fourier descriptor
HSV	Hue-saturation-value
RGB	Red-green-blue
RKB	Red Kidney Bean
SVM	Support Vector Machine

List of Figures

Figure 2-3-1: The process of converting analog image into digital image	10
Figure 2-4-1: Sobel kernels with 3 x 3	14
Figure 2-4-2: Architecture of a Backpropagation Neural Network	17
Figure 3-2-1: The system architecture for the proposed system.....	26
Figure 3-4-1: The pre-processing flowchart	28
Figure 3-4-2: “A” False Regions Shown as Tiny White Spots in the Background “B” RKB Kernels after False Region Removal	28
Figure 3-5-1: The proposed flowchart for image segmentation	29
Figure 3-5-2: “A” Original image, “B” Result of Thresholding	30
Figure 3-5-3: RKB Image and its RGB Components	31
Figure 3-5-4: Reconstructed images of the Red, the Green and the Blue Binary Images	32
Figure 3-6-1: Screenshot Showing Values of 14 Color Features for 12 Healthy RKB Kernels...	33
Figure 3-6-2: Screenshot of the Values of 2 Size Features for 12 Discolored RKB Kernels	36
Figure 3-6-3: Screenshot of the Values of 8 Shape Features for 12 Healthy RKB Kernels	37
Figure 3-6-4: Shape Features Extraction Algorithm	37
Figure 3-7-1: Design of ANN Used for the Classification of RKB Sample	39
Figure 4-3-1: Result of threshold segmentation algorithm at different thresh levels	45
Figure 4-3-2: Sobel edge detection algorithm	46
Figure 4-4-1: Graphical user interfaces	48
Figure 4-4-2: Cross-Entropy Error Showing the Performance of the Trained ANN	50
Figure 4-5-1: “A” Spots formed by healthy; “B” Spots formed by insect board	54
Figure 4-5-2: Comparison of Discriminative Power of Size, Shape, and Colour features	55

List of Tables

Table 2-5-1: Summary of related works	21
Table 3-6-1: Color Feature Values for Samples of Healthy and Defect Kernels	35
Table 3-9-1: The Confusion Matrix of a Classifier with Two Classes	41
Table 4-2-1: Data set description	44
Table 4-4-1: Test Confusion Matrix of Naïve Bayesian Classifier	49
Table 4-4-2: Confusion Matrix Showing Overall Classification Accuracy	50
Table 4-4-3: Performance of model in different classifiers	51
Table 4-4-4: Confusion Matrix for Scenario-1	52
Table 4-4-5: Confusion Matrix for Scenario-2	52
Table 4-4-6: Confusion Matrix for Scenario-3	53
Table 4-5-1: The four scenarios discriminative efficiency	54

Abstract

The red kidney bean (RKB) is a vital crop whose distribution in the market is subject to stringent quality control. RKB samples are now manually evaluated using ocular inspection, with the contents classified as foreign matter, defect, healthy, contrast, and insect board kernels. Visual examination, on the other hand, necessitates a significant amount of time as well as the presence of qualified and experienced professionals. Furthermore, it is influenced by human nature's biases and inconsistencies. Such a procedure cannot be adequate for large-scale examination and grading unless it is fully automated.

The goal of this study is to create a system that can evaluate the quality of RKB sample elements utilizing digital image processing techniques, RKB image data is collected from ECX warehouse, the sample of RKB providing a total of 62 samples, which yielded 582 sample images. Image preprocessing are the steps taken for the improvement of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis task then a novel segmentation technique is proposed to segment the foreground from the background, partitioning both RKB and foreign particles and lay the foundation for feature extraction. To model RKB sample ingredients, a total of 24 features (14 colors, 8 shapes, and 2 sizes) have been extracted. The data set is randomly apportioned into training and test set with 70% and 30% proportion, respectively. Classification algorithms, such as artificial neural networks and naïve bayes classifiers are applied based on the Ethiopian Commodity Exchange (ECX) RKB standard.

Using a feed-forward artificial neural network classifier with a back propagation learning algorithm, 24 input nodes, and 5 output nodes, matching the number of features and classes, has been constructed for the classification of RKB samples. Accordingly, the classifier achieved an overall classification accuracy of 93.8%. The success rates for detecting foreign object, defect, healthy, insect board, contrast, kernels are 100%, 92%, 95.2%, 84.4% and 100% respectively. This research work does not include moisture content analysis of RKB. It is therefore recommended as a future research direction to enhance the performance of the proposed model in this study.

Keywords: RKBs quality assessment, Image segmentation, Digital image processing, Classification algorithms

CHAPTER ONE

Introduction

1.1 Background

Red kidney bean is one of the most significant grain legumes grown in Ethiopia's lowlands, especially in the Rift Valley [1]. RKB is grown in these areas for both export and home use. Red kidney bean is also a major food crop in Ethiopia, notably in the south and east. The planting period for red kidney beans should be carefully selected so that harvest occurs during the dry season or before the commencement of rainy periods with a reasonable rainfall of 450-700mm. The usual time from planting to harvesting is 95-100 days [1].

Red kidney bean is a highly exported crop in a large amount. In 2005/06 for instance, Ethiopia exported about 62,262 tons of RKB valued at about 22 million USD or about 193.7 million ETB, with a unit value of export of 353USD/mt [1]. Red kidney bean trading and circulation in Ethiopia are subjected to a standard set by the Ethiopian Standards Agency (ESA). This standard sets criteria by which Red kidney bean quality is evaluated. The standard is based on the morphological and color characteristics of RKB. Currently, red kidney bean quality prediction is assessed manually. However, manual evaluation takes a significant amount of time and requires trained experts [1, 2]. Thus, considerable emphasis should be placed on keeping the accuracy of the grading technique to maintain the quality of red kidney bean grains. Thus, a better way to control the quality and screen out the unwanted product effectively, an automated system of control is needed. Digital image processing is playing a big role in controlling and assessing the quality of agricultural products [2].

Image processing is a method to perform some operations on an image, to get an enhanced image, or to extract some useful information from it [61]. It is a type of signal processing in which input is an image and output may be an image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms a core research area within engineering and computer science disciplines [3].

The basic steps of Image processing include the following [3]. The first step is Image Acquisition during which the camera is used for capturing grain images in digital form and stores them in any digital media. This is followed by image pre-processing which removes noise, smoothens the image, and also performs resizing of images. RGB images are converted to grey images, in addition to increasing the contrast of an image at a certain level. The third step is image segmentation, which is used for partitioning an image into various parts. Image segmentation is followed by feature extraction, for obtaining features like color, texture, and shape which reduce resources to describe a large set of data before classification of the image. Finally, Classification algorithms can be employed to analyze the numerical property of image features and organize its data into categories.

1.2 Motivation

Technological advancement is gradually finding applications in the agricultural and food industries, in response to one of the greatest challenges; i.e. meeting the need of the growing population. Efforts are being geared towards the replacement of human operators with automated systems, as human operations are inconsistent, prone to error, and less efficient. Automation means every action that is needed to control a process at optimum efficiency as controlled by a system that operates using instructions that have been programmed into it or response to some activities. Automated systems in most cases are faster and more precise [5].

With the progress in computer image vision technology, the gradation technique based on computer vision has developed. The computer vision gradation technology is real-time, objective, non-destructive, and can detect multi-index simultaneously, such as size, defection, color, shape, and maturity [6].

In Ethiopia, technologies of image analysis or computer vision have not been explored in a significant manner in the development of automation in the agricultural and food industries [7]. Specifically, Ethiopian red kidney bean quality inspection is based on the traditional conduct of the grading system. This manual grading evaluation by visual inspection is labor-intensive, time-consuming, and suffers from the problem of inconsistency and inaccuracy in judgment by different perceptions of humans. Thus, we should automate the grading system of red kidney bean agricultural products. Automated red kidney bean gradation system plays a significant role to increase the value of produces. Commonly, the gradation indices are shape, size, color, maturity, defection, etc. [8].

Automated grading and sorting of agricultural products are getting special interest because of increased demand for different quality food with relatively affordable prices by the different groups of customers belonging to different living standards [9]. Therefore, the operation of imaging technology in the area will have great importance to enable profitable activities by increasing efficiency and promoting the market.

1.3 Statement of the Problem

Ethiopia is a developing country where agriculture is the backbone of the economy and the source of livelihood for 84% of the population [10]. The ECX was established to modernize the Ethiopian agricultural market and transform the economy through a dynamic, efficient, and transparent marketing system.

Hence, maintaining the quality of agricultural products, including red kidney bean grain is the main goal of ECX members and experts. However, most of the exported red kidney bean products have been facing several quality degradation problems. One of these problems is due to the use of traditional grading techniques and the lack of advanced measuring equipment.

Nowadays Ethiopian red kidney bean grain classification and grading are performed manually. As stated earlier, this technique has its drawbacks, such as being prone to error, labor-intensive,

inconsistent, aged, and cost-ineffective. Nowadays, this has become a big issue for the Ethiopian Commodity Exchange (ECX) because it costs a lot of money and also causes to loss of its reputation. In other words, it loses its market share in the international market. Thus, to be competitive in the market and proliferate its market share, it needs to improve the existing grading technique to compete with other countries. This intrigue to come up with an automatic way of grading the red kidney bean based on their physical appearance [2]. In line with this, the exactness of quality scrutiny via a human assessment scheme is different from person to person according to the inspectors' physical status such as working hassle, point of view, and fidelity for traders. In general, manual sorting, grading, and classification which are based on a traditional visual quality inspection performed by human operators are tedious, time-consuming, slow, and consistent [2].

Few researchers have conducted an automated classification and grading system for different agricultural products such as sesame [11], white pea beans [10], Maize [12], and coffee bean [13]. Hiwot [11] shows that the classification and grading technique proposed for an agricultural product will not be directly applied for others due to the difference in morphological, color, and texture features. This happens because of differences in seed shape, size, color, and other characteristics of the product.

To the best of our knowledge, there is no prior work attempting to develop a system for the classification and grading of Ethiopian red kidney bean grain. Thus, this research work aims to explore and develop an automatic red kidney bean grain classification and grading system taking the physical characteristics into account.

Quality prediction system using machine learning for red kidney beans is relevant to explore the possibilities of adopting a faster system that saves more precise in the grading of red kidney beans by reducing experts effect of bias related to the quality standard that improves the commercial needs. Hence, replacing manual inspection of red kidney beans with a superior speed, a precise, cost-effective, consistent, non-destructive automated system is necessary for red kidney beans in which it generates a huge amount of income and forges currency to the country.

It is therefore the aim of this study to design and develop a system for red kidney bean classification and grading using digital image processing and machine learning approaches.

To this end, this study attempts to investigate and answer the following research questions.

- What are the suitable preprocessing algorithms to remove noise introduced in sample RKB images?
- Which segmentation algorithm is suitable to isolate RKB from the background and separate the connected RKBs?
- What are the suitable features for the classification of RKB?
- To what extent the proposed model works for the classification & grading of RKBs?

1.4 Objective of the Study

1.4.1 General Objective

The general objective of this study is to propose a model for Red Kidney Bean classification and grading using digital image processing and machine learning techniques

1.4.2 Specific Objective

The specific objectives of this research are the following.

- To review the literature to identify methods and algorithms used for applying image processing
- To collect and prepare different samples of RKB images for training and testing
- To select appropriate preprocessing and segmentation algorithms to analyze the images of RKB
- To identify and analyze different features of RKBs that are useful for the prediction and classification of RKB
- To develop the prototype for the classification and grading of RKB
- To test and evaluate the performance of the proposed prototype

1.5 Scope and Limitation of the Study

The purpose of this thesis work is to automate the classification & grading of RKB using the approaches of image processing techniques.

Generally, this research work is grounded on the physical property of RKB that are characterized as morphological features and color, shape, foreign matter, and texture characteristics. It does not include moisture content analysis mass determination and chemical content analysis of RKB.

The source of data for red kidney beans is the ECX warehouse and the type of data considered for the study is an image of RKB. The labels of the image taken include Healthy, Defect, Contrast, Foreign object, and Insect board. In addition, the image is used at the stage of experimenting following the image processing steps, such as preprocessing, segmentation, feature extraction, classification, and grading. Computational resources (processor, RAM) required for processing the sample constituents of the image plus the time it takes to process is another limitation of the study.

1.6 Significance of the Research

This finding of the study will have a great benefit for ECX, agricultural experts, and researchers, as presented below.

- It will minimize the processing time and labor cost. This will also improve the quality-based export of red kidney bean grain.

- It gives a platform to conduct grading at one specific place, centralization. This in turn will enable ECX to have the same standard across all products and quality control will be easy.
- It also helps to minimize corruption that might arise due to manual grading so that the exporter or merchants may corrupt the grading experts.
- To reduce capabilities of decision-making that comes from human inspector physical conditions such as fatigue and eye sight, a mental state caused by biases and work pressure, and working conditions such as improper lighting, climate, etc.
- It will benefit researchers who need to take part in achieving the goal of developing efficient digital image processing techniques for different agricultural products.

1.7 Methodology

The focal issue of this study is the development of a machine learning vision system aiming at the establishment of technological and innovative approaches towards sample red kidney bean raw quality value classification by extracting the relevant red kidney bean features. This study is therefore conducted to have a better understanding of the existing problem and provide results that have a great contribution to improving the quality of red kidney bean grain.

1.7.1 Research Design

This study follows experimental research. Experimental research seeks to determine a relationship between two variables; the dependent and the independent variables. After completing an experimental research study, a correlation between a specific aspect of an entity and the variable being studied is identified [14].

This kind of research involves data preparation, selecting implementation tools to experiment algorithms, and finally evaluation of the proposed system.

1.7.2 Data Collection & Preparation

Red kidney beans are produced in different regions of the country and there are multiple distribution points throughout the country. Sample data were acquired from the ECX warehouse in Addis Ababa.

Sampling is one of the main procedures used in the current manual system for experimenting with red kidney beans quality classification and evaluation. In the current manual system practice, the sample drawer draws 3kg of "representative" samples for every 10 tons from the truck, which is the average load capacity of the truck when it arrives. Of the 3 kg, 125 g is used for analysis, and the rest is used for other references. In this regard, we took 62 images, which contained 582 RKB images? A digital camera, Huawei Model TRT-L21A with a specification of 12.1 megapixels, was used to capture red kidney bean kernel images. Out of these samples, 70% are used for training and 30% for testing and verification purposes. The red kidney bean samples came from the Ethiopian Commodity Exchange warehouse (ECX).

Then preprocessing is performed to remove noise and enhance image quality and different feature information is extracted from each red kidney bean. This is a pre-processing activity.

1.7.3 Implementation Tools

Towards classifying and grading RKBs, it's a prototype is developed using MATLAB version R2018a. MATLAB is a software developed by MathWorks. Matlab began as a straightforward "Matrix Laboratory." J. H. Wilkinson, George Forsythe, and John Todd were three men who performed significant contributions in the development of MATLAB, which now supports a wide range of techniques in Computer Vision and Machine Learning. MATLAB is the easiest and most productive computing environment for engineers and scientists. MATLAB is one of the well top programming languages dedicated to mathematical and technical computing. MATLAB is selected for this study since the researcher has better know-how to use this language for coding and developing the prototype for experimentation. Millions of engineers and scientists worldwide use MATLAB for a range of applications, in industry and academia, including deep learning and machine learning, signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. Image processing in MATLAB involves processing an image into fundamental components to extract meaningful information. Image analysis can include tasks such as finding shapes, detecting edges, removing noise, counting objects, and calculating statistics for texture analysis or image quality. MATLAB also makes machine learning easy. With tools and functions for handling big data, as well as apps to make machine learning accessible, MATLAB is an ideal environment for applying machine learning to image data analytics [60].

Visio 2013 was used for designing the proposed architecture & model different kind's figures in this study, Visio it easy to design any kind of model without a need for pre-training, there are also several packages with a variety of model types corresponding to different knowledge areas, in addition, it's a Microsoft product which is easily integrated and work with Microsoft word.

1.7.4 Evaluation Method

Evaluate each model by running the test data set on the classifier created using the training data set. The performance of the classifier model is returned as the result, which contains the performance matrix and percentage accuracy metric for each category, which is summarized in the confusion matrix. The confusion matrix is a contingency table used to generate true positive (TP), true negative (TN), as well as false positive (FP), and false-negative (FN). TP and TN indicate the correct allocation of samples into their respective categories, whereas FP and FN indicate the wrong classification of samples from their respective categories [56].

Based on the confusion matrix, the common effectiveness metrics such as accuracy, recall, and precision are used to measure overall and each class performance. Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations, While Precision is the ratio of correctly predicted positive observations to the total

predicted positive observations, Recall is the ratio of correctly predicted positive observations to total observations in actual class [56].

1.8 Organization of the Thesis

The rest of this thesis is organized into five chapters. Chapter one provides an introductory, statement of the study, the objective of the study, scope & limitation, significance of the study, methodology, and research design.

In Chapter two, literature will be reviewed concerning pattern recognition techniques for identifying and classifying cereal grains, classification of olive oil seeds using ANN classifier, a method for the classification of cereal grains, namely; barley, rye, oats, and wheat, the new algorithm used to extract different features of bulk grain samples by the using of the neural network, a system for the automatic analysis of Heliiothiszea insect by using of image analysis, in addition, related works are reviewed to pinpoint the research gap.

Chapter three gives a detailed description of the methodology of the thesis is including the proposed architecture, image data collection image preprocessing, segmentation, feature extraction, classification algorithms, evaluation method, and summary.

Chapter four, experimentation of the study presents the dataset & their corresponding classes, experiments segmentation, a test result of the classification algorithm, selecting an optimal model for classification and grading of RKBs, discussion of result then finally contribution knowledge is discussed.

In Chapter five conclusions of the thesis will be drawn and future works will be pointed out.

CHAPTER TWO

Literature Review

2.1 Overview

In the late 1960s and early 1970s, image processing techniques were first applied in medical imaging, remote earth resource observation, and astronomy. Digital image processing has advanced quickly since then. Techniques for digital image processing are currently being employed in a range of other applications. Digital image processing is increasingly employed in a variety of applications, including product inspection, robotic vision, and scientific research [16].

This chapter examines the types of literature connected to the notions that constitute the foundation of this thesis. We begin by presenting a summary of Ethiopian red kidney beans, followed by the ECX's current grading techniques. Then, numerous image processing techniques are comprehensively addressed, including picture acquisition, preprocessing, and several forms of segmentation, feature extraction, and classification. Finally, related works are scrutinized in order to discover the gap that this study intends to fill.

2.2 Red kidney Bean Sample Constituents

In this work, an attempt is made to do red kidney bean classification and grading based on bean photographs. The following are descriptions of the elements of red kidney bean (RKB) samples [17].

- **Foreign matters-** other than RKB, foreign matters include all organic and inorganic things such as plant pieces, sand, soil, glass, and filth.
- **Moisture content**– the amount of water recorded in a pulse sample that represents a load of pulses tendered for delivery.
- **Insect bored-** are kernels with prominent weevil-bored holes or evidence of boring or tunneling, showing the presence of insects, insect webbing, or insect refuse grains eaten in one or more parts of the kernel, exhibiting obvious traces of a vermin attack.
- **Defect -** RKB grains have a thin, papery look and are undeveloped.
- **WSD -** are grains that become unfit for human food as a result of rot, mold, bacterial decomposition, or other factors that can be detected without having to cut the grains to inspect them.
- **Contrast color** – A seed coat or kernel that is noticeably different in color from the predominant hue of the specified material's predominating class.
- **Healthy -** RKB kernels are normal in size and have no damage.

2.3 Digital Image Representation

There are different picture acquisition methods, but the end purpose is the same: acquire digital images from sensed data. The output of most sensors is a continuous voltage waveform. The amplitude and spatial behavior of these waves are related to the natural phenomenon being observed. As a result, to create a digital image, continuously detected data must be transformed into the digital form [16, 18].

As a result, a digital picture may be thought of as a discrete data representation that comprises both spatial (layout) and intensity (color) information. A continuous image is continuous in both the spatial and the amplitude domains. We must sample the function's coordinates and amplitudes in order to convert it to digital form. The process of digitizing coordinate values is known as sampling. The process of digitizing amplitude values is known as quantization [16].

Consider $f(s, t)$ to be a continuous image function of two continuous variables, s , and t , to better comprehend sampling and quantization. This function generates a digital image. Assume we sample the continuous image into a 2-D array, $f(x, y)$, with M rows and N columns and discrete coordinates (x, y) . For notational clarity and simplicity, we employ integer values for these discrete coordinates: $x = 0, 1, 2, \dots, M - 1$ and $y = 0, 1, 2, \dots, N - 1$. As an example, the value of the digital image at the origin is $f(0, 0)$, and the following coordinate value down the first row is $f(0, 1)$. This does not necessarily indicate that these were the physical coordinate values at the moment the image was sampled. $f(x, y)$ denotes the image value at any coordinate (x, y) , where x and y are integers. The spatial domain is the piece of the real plane encompassed by an image's coordinates, and x and y are known as spatial variables or spatial coordinates [16, 19]. The sampling and quantization transformation process are depicted in general in Figure 2-3-1.

There are two key ways to represent $f(x, y)$. The first method is to plot the function, with two axes determining the spatial location and a third axis containing the values of f , also known as intensities as a function of the two spatial variables x and y . Complex images, on the other hand, are typically too detailed and difficult to interpret from such plots. When working with gray-scale sets, this representation is useful because the elements are expressed as triplets of the form (x, y, z) , where x and y are spatial coordinates and z is the value of f at coordinates (x, y) [16, 18].

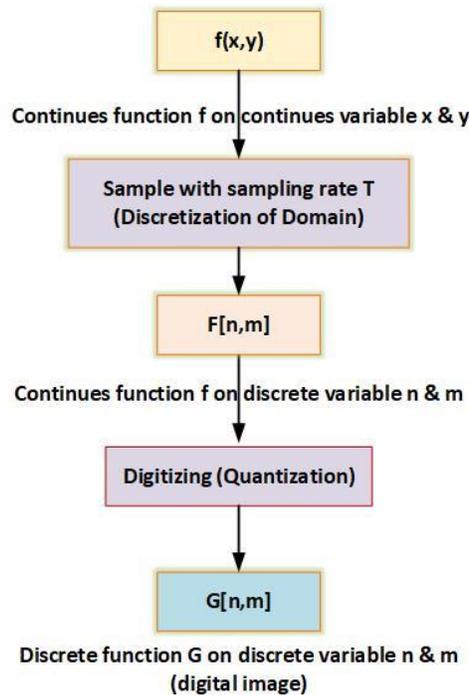


Figure 2-3-1: The process of converting an analog image into a digital image [16]

The second method simply displays the numerical values of $f(x, y)$ as an array known as a matrix. This format is useful while developing algorithms. As illustrated in figure 2-3-1 [16], we express the representation of a $M \times N$ numerical array in equation form.

$$f(x, y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1,0) & f(M - 1,1) & \dots & f(M - 1, N - 1) \end{pmatrix}$$

Both sides of this equation are quantitative representations of a digital image. The right side is a matrix of real numbers. The members of this matrix are known as image elements, picture elements, or pixels [16].

The position of the (x,y) plane's origin, as well as the positive x and y axes' directions, are critical during digital picture representation. As a result, the origin of a digital image is located in the upper left corner. Furthermore, the positive x -axis extends below while the positive y -axis extends to the right. This is a frequent representation because many image displays, such as television monitors, sweep an image from top left to right one row at a time. As a result, because the first element of a matrix is always at the top left of the array, locating the origin of $f(x, y)$ there makes mathematical sense [16, 18, 19].

2.4 Digital Image Processing

The field of digital image processing refers to the use of a digital computer to process digital images. It entails editing or modifying an existing image in the desired way. Image processing can be divided into three high-level steps or processes. These stages are referred to as low-level, mid-level, and high-level processes [16].

Primitive operations such as image preprocessing to remove noise, contrast enhancement, and image sharpening are examples of low-level processes. A low-level process is distinguished by the fact that its inputs and outputs are both images. Image segmentation is an example of mid-level processing. This is the task of describing those objects in order to convert them to a form that can be processed by a computer, as well as the classification of individual objects. A mid-level process, as opposed to low-level processing, is distinguished by the fact that its inputs are typically images, while its outputs are properties retrieved from those images. Edges, curves, and the individuality of particular objects are examples of these qualities. Finally, higher-level processing entails "making sense" of a collection of recognized things [16]. These image processing steps are explored further below.

2.4.1 Image Preprocessing

One of the low-level operations in image processing is picture preprocessing. We may end up with noisy photos as a result of image acquisition. Image noise can come from a variety of sources [18]. The precise focus of the camera is critical during image acquisition. As a result, if the camera is not correctly focused, we obtain blurry photographs.

Other factors can contribute to the existence of noise in images. Noise in images can be introduced by conditions such as a cloudy atmosphere and relative motion between the object and the camera. As a result, if the camera is pushed during the image capture interval while the object is immobile, the resulting image will invariably be blurry and noisy. Noise has a noticeable negative impact on digital picture processing. There are methods for dealing with noise in digital image processing [18].

Re-Sampling: This is the process of converting a sampled image from one coordinate system to another (changing an image's pixel dimension) [53]. The display size of an image is also affected by re-sampling. When we resample an image, we either lose part of the image's information (down sampling) or increase the number of pixels in the image; new pixels are added based on the color values of existing pixels.

Image Enhancement: One of the simplest and most appealing fields of digital image processing is image enhancement. The goal of enhancement techniques is to bring out detail that has been obscured in an image, or merely to highlight certain features of interest. When we raise the contrast of an image to make it look better, this is an example of enhancement. It is critical to remember that picture enhancement is a highly subjective area of image processing.

Image Restoration: Image restoration is the process of enhancing the appearance of an image. Image restoration, unlike augmentation, is objective in the sense that it is based on mathematical or probabilistic models of image degradation.

2.4.2 Image Segmentation

The technique of dividing an image into regions or objects is known as image segmentation. It is a middle-level image processing phase. It is the first step in the picture analysis process. Picture segmentation is the division or separation of an image into sections with comparable attributes [25]. As a result, the fundamental notion behind image segmentation is to group individual pixels into areas if they are comparable. Similar can indicate they have the same intensity (gray shade), make a texture, line up in a row and form a shape, and so on. As a result, pixels in a region have similarities based on homogeneity criteria such as color, intensity, or texture, which can be used to find and identify objects and borders in an image [14, 15, 16]. The following is a description of picture segmentation, as presented in [18]. A complete segmentation of an image R entails identifying a limited number of regions ($R_1, R_2, R_3, \dots, R_N$) in such a way that [25]:

- i. $R = R_1 \cup R_2 \cup \dots \cup R_N$ – The union of all the sub-regions gives the original region
- ii. $R_i \cap R_j = \emptyset \quad j \neq i \forall$ – The sub-regions don't have an intersection

The algorithms for segmentation are based on one of the two essential features of gray-level values. One is based on gray-level value discontinuity, while the other is based on gray-level value similarity [25].

We segment a picture in the gray level values discontinuities based on abrupt changes in the gray level. The identification of lines and edges in an image is of particular relevance in this category. Thus, if we can extract and link the edges in an image, the region is defined by the edge contour that contains it. In this sense, the connected sets of pixels have about the same homogenous intensity as the regions. Thus, the pixels within the areas describe the region, and the segmentation procedure entails partitioning the entire scene into a finite number of regions. The second method is to compare the gray levels. It is determined by the similarity of pixels inside a region. Various local features of the pixels are used when segmenting an image. There are various sorts of well-known segmentation techniques. Among these, histogram-based thresholding and edge detection will be discussed here [18].

2.4.2.1 Histogram Based Thresholding

The thresholding operation entails identifying a set of appropriate thresholds, which are then used to split the image into numerous meaningful areas [18]. To threshold, a grey level image, compare each pixel value to a predefined number known as a threshold. The first stage in image segmentation is to separate the foreground from the background of a picture by changing it from a color image to a binary image with just 0 and 255 potential pixel values. The value of the background pixels could be 0 and the value of the foreground pixels could be 255, or vice versa.

Each pixel is inspected during thresholding to identify whether it belongs in the background or the foreground. This is accomplished by comparing each pixel value to a fixed value known as

the threshold. As a result, if the pixel value is less than the constant, it is set to 0; otherwise, it is set to 255 [16, 18, 19]. Unfortunately, because greyscale images are one-dimensional, the above explanation works well. RGB images, on the other hand, are made up of three color bands [16]. These bands of color are known as red, green, and blue, in that order. When RGB photos are converted to grayscale, color information is lost, which is critical for identifying red kidney bean kernels. As a result, in addition to the selection of threshold values, RGB color image thresholding necessitates the selection of one of these three color bands.

Thresholding is a straightforward but effective method for segmenting photographs with luminous objects on a dark backdrop [20]. Thus, gray level thresholding is based on an examination of an image's histograms. The amount of peak values in the histogram influences its analysis.

2.4.2.2 Edge-Based Segmentation

Edge detection algorithms can also be used to segment data. To identify visual discontinuities, edges are recognized. The region's edges are identified by recognizing the pixel value and comparing it to surrounding pixels. Pixels that do not share an edge are assigned to the same category. When the gray levels of the items vary, darker objects become too small, and brighter objects get too huge. The size fluctuations are caused by the fact that the gray values at an object's edge move gradually from the background to the object value. If we use the mean of the object and the background gray values as the threshold, there is no size bias. This method, however, is only practicable if all items have the same gray value or if we use distinct criteria for each object. As a result, edge-based segmentation is based on the fact that the position of an edge is supplied by a first-order derivative extreme or a zero crossing in the second-order derivative [21, 22].

To segment the image, various edge detectors are utilized. The technique used to segment images is known as differential operators. The differential operator is a traditional edge detection approach that is based on the gray change of a picture for each pixel in its area. Convolution is used to do this. One of the first-order differential operators is the Sobel edge detector [23]. The Sobel edge detection procedure extracts all edges in an image, independent of their direction. It is realized as a collection of two-directional edge improvement techniques. The generated image is a unidirectional outline of the source image's objects.

Constant brightness parts are darkened, whilst changing brightness regions are emphasized. The derivative can be implemented digitally in a variety of ways. The sobel operators, on the other hand, have the advantage of delivering both a differencing and a smoothing effect. Because derivatives amplify noise, the smoothing effect of the sobel operators is particularly appealing [23].

-1	0	1
-2	0	2
-1	0	1

G(x)

1	2	1
0	0	0
-1	-2	-1

G(y)

Figure 2-4-1: Sobel kernels with 3 x 3 [21]

2.4.3 Feature Extraction

A distinguishing primitive property or attribute of an image is referred to as an image feature. One of the most important aspects of image analysis is the extraction of sufficient information to produce a concise description of an analyzed image [12]. Because of the enormous size of digital photographs, analyzing an image in its original form can be time-consuming. Some quantitative information is collected from the items to be evaluated in the image to make the image analysis process simpler and less time-consuming. The computational cost of object recognition is considerably decreased by isolating the region of interest, enhancing recognition efficiency [24]. Image features are extremely important in image classification. For picture categorization, several types of image features have been proposed. Some of the fundamental visual features are morphology, color, and texture [18, 25, 26].

Morphological features, such as shape and size, are geometric properties of a picture. They are physical dimensions metrics that characterize an object's appearance. For example, area and perimeter are two of the most regularly assessed size characteristics, and circularity assesses the shape of picture compactness [26].

Morphological traits are commonly employed in the industry for automated grading, sorting, and detection of objects [26]. In some applications, such as cereal grain categorization, these features alone are insufficient for a high-performance inspection process and must be coupled with others. The attributes of pixels within the object boundary are used to extract color and textural details [24, 27, 28, 29, 30, 31, 32, 33].

Color, along with geometrical elements, is one of the most commonly utilized features for image classification. Each pixel in an image records a numeric value, which is frequently the brightness of the associated place in the image. Color information can be represented by combining several such values. The most common brightness value range is 0 to 255 (8-bit range), although depending on the type of camera, scanner, or other acquisition devices, a broad range of 10 or more bits, maybe up to 16 (0 to 65,535), may be encountered. However, because such arrays are easier to operate and convert to displays, most images are still stored using a collection of discrete integer grey values [26]. Accordingly, statistical values of color features such as mean, mode, standard deviation, and so on are employed for image categorization.

Several features are used to assess the morphological, color, and shape characteristics of objects under inquiry [26]. Kernel width (minor axis length), kernel length (major axis length), area,

perimeter, color, aspect ratio, ovality, solidity, and convexity are the most critical [34]. Kernel length is defined as the greatest distance between the kernel's farthest ends. Some sources refer to kernel length as major axis length, whereas kernel width is the longest line that can be drawn perpendicular to the kernel length through the item. Kernel width is referred to as minor axis length in some research [35, 36, 38].

The number of pixels included within a kernel's perimeter is defined as its area. Kernel area is calculated by counting the total number of pixels in the binary picture that corresponds to the item. We can compute the perimeter of the kernel by walking around it pixel by pixel.

A kernel's perimeter is the length of its boundary. In other works, this parameter is referred to as circumference [35, 36, 38]. Aspect ratio, one of the most frequent shape feature descriptors, is calculated by dividing the main axis length by the minor axis length. This metric is referred to as elongation in [37]. Similarly, ovality, solidity, and convexity are calculated as a ratio of two measurements. Ovality is defined as the ratio of an object's area to the area of an ellipse with the same major and minor axes as the object. The solidity of the kernel is calculated by dividing the kernel area by the area of the convex hull. The convex hull is the smallest convex polygon that can contain the kernel seed region in this situation. Convexity is another parameter that makes use of the concept of a convex hull. It is the ratio of the kernel's perimeter to the convex-hull polygon's perimeter [35, 36, 38]. As a result, visual attributes such as morphology, color, and texture are fed into a pattern classifier, which categorizes objects, in this case, red kidney beans, into separate groups.

2.4.4 Classification

The process of finding a model that defines and distinguishes data classes or concepts is known as classification [39]. These models are known as classifiers, and their goal is to predict categorical class labels. Classification is a two-part procedure that includes both a learning and a classification stage. A classification algorithm creates the classifier in the learning phase by analyzing or "learning" from a training set made up of database tuples and their associated class labels, whereas the model is used to predict class labels for given data in the classification step [39].

The resultant model can be expressed in a variety of ways, including classification rules, often known as IF-THEN rules, decision trees, mathematical equations, or neural networks. A decision tree is a tree structure that looks like a flowchart. A decision tree is made up of nodes, branches, and leaves. Each node represents a test on an attribute value, each branch represents a test result, and the tree leaves represent classes or class distributions. Classification rules can be simply converted from decision trees. A neural network, on the other hand, is often a collection of neuron-like processing units with weighted connections between the units when used for classification [39].

a) Naive Bayesian classifier

The Naive Bayesian classifier is based on probability distributions. Based on the observable features, it classifies an object into the class to which it is most likely to belong. It is the result of using the Bayes Theorem with independent assumptions about the features. Simply put, a Naive Bayesian classifier believes that the value of one feature is unrelated to the presence or absence of any other characteristic. It performs well when the training data does not include all possible outcomes, allowing it to be particularly effective with small quantities of data [40]. The Bayesian classification method is discussed in detail in [40].

Assume that in a d-dimensional feature space $X = (X_1, X_2, \dots, X_n)$, there are N classes C_1, C_2, \dots, C_N and an unfamiliar pattern x . Calculate the likelihood of the pattern X belonging to each class $C_i, I = 1, 2, \dots, N$. If the chance of the pattern belonging to C_k is the greatest, it is classified as belonging to C_k . When using Bayesian classification to classify a pattern, we distinguish between two types of probabilities. There are two types of probability: priori probability and posteriori probability. The priori probability expresses the likelihood that the pattern would fit into a class, say C_k , based on prior belief, evidence, or knowledge.

The posteriori probability $P(C_i/x)$ on the other hand, denotes the final probability of the pattern x belonging to a class C_i . The posteriori probability is calculated using the pattern's feature vector, class conditional probability density functions $P(x/C_i)$ for each class C_i , and the priori probability $P(C_i)$ for each class C_i .

According to Bayesian classification, the posteriori probability of a pattern belonging to a class C_k is given by, C_i [40]:

$$P\left(\frac{C_k}{x}\right) = \frac{P\left(\frac{x}{C_k}\right) P(C_k)}{\sum_{i=1}^N \left(P\left(\frac{x}{C_i}\right) P(C_i)\right)}$$

Where $(P(x / C_i) * P(C_i))_{i=1}^N$ denotes the posteriori probability that the pattern x belongs to class C_i .

c) ANN Classifier

Instead of programming a computational system to do a specific task, a neural network model, a branch of artificial intelligence, teaches the system to execute tasks. It is composed of many artificial neurons that are linked together using an explicit network architecture. The neural network's goal is to receive one or more inputs and sum them to produce an output. Typically, the sums of each node are weighted, and the sum is passed through an activation or transfer function. The mode of instruction can be either supervised or unsupervised. ANNs have the potential to solve problems where the inputs and output values are known but the relationship between the inputs and outputs is difficult to translate into a mathematical function. It predicts when patterns are too complex for humans or other computer techniques to detect [41, 42]. Feed-forward backpropagation (B-P) is the most commonly used ANN classifier (see the below figure 2.4).

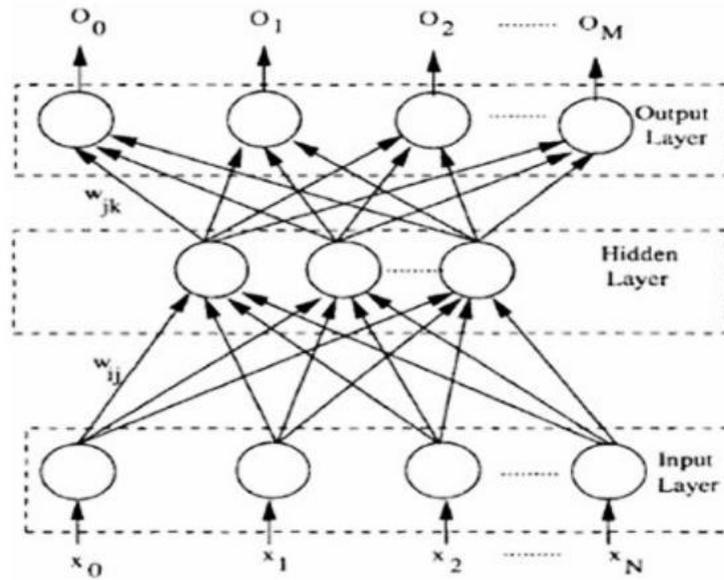


Figure 2-4-2: Architecture of a Backpropagation Neural Network [41]

The Feedforward B-P Algorithm

To train the neural network, it goes through two distinct passes: a forward pass (computation of all the neurons' outputs) followed by a backward pass (error propagation and weight adjustment) through the network's layers [41]. As it scans the training data, the algorithm alternates between these passes several times. Thus, with the proper combination of training, learning, and transfer function, the dataset classification employs the most effective tool known as back propagation neural network [41].

The sigmoid logistic function used by a standard back-propagation algorithm can be generalized to[41], where parameters such as K (Killback-Liebler information distance), Learning with logarithmic error metrics was also less prone, and the parameter D (sharpness or slope) of the sigmoidal transfer function are considered to measure the network's efficiency.

$$f(\xi) = \frac{K}{1 + \exp(-D \cdot \xi)} - L$$

2.5 Related Works

Accurate classification, grading, and sorting of foods or agricultural products are required to increase expectations in terms of food quality and safety standards. Thus, computer vision and image processing were nondestructive, accurate, and dependable methods for achieving the goal of sample product classification and grading.

As a result, in order to address this issue, we considered some related works. This chapter discusses grain classification and quality assessment.

Zayas et al. [43] illustrated the use of image analysis to distinguish between wheat and non-wheat components in a grain sample. They presented two pattern recognition methods: multivariate discriminate and a structural prototype method. The main issue with this method is that irregularly shaped stones are misclassified as wheat. The accuracy of quality classification is about 90%, and the accuracy of grading is about 92 percent. The proposed method has a limitation in that the kernels must be manually oriented.

Lai et al. [44] proposed pattern recognition techniques for identifying and classifying cereal grains. For the samples used in the study, this method produced 100 percent accurate predictions. The pattern obtained was chosen from a large number of possibilities. These are obtained through subjective judgment and trial and error. Corn, wheat, soya bean, and sorghum were the grains considered here. This study's findings illustrate and document general approaches to image analysis and techniques that can be used to characterize cereal grains. The selection of variables to create a reliable pattern to characterize and distinguish specific grain from others was a major limitation.

Visen et al. [45] proposed color image acquisition and processing algorithms for five-grain types: oats, barley, rye, wheat, and durum wheat. Over 150 color and textural features were extracted using the developed algorithms. To identify the unknown grain types, a back propagation neural network-based classifier was developed. For training purposes, the neural network was shown color and textural features. After that, the trained network was used to identify the unknown grain types. Classification accuracies of more than 98 percent were obtained for all grain types; however, the accuracy level obtained for oats is significantly lower than for the other grain types.

Anami et al. [46] used a Neural network approach to classify single grain kernels of various grains such as wheat, maize, groundnut, red gram, green gram, and black gram based on color, the area covered, height, and width. When food grain image samples are in their purest forms, the maximum and minimum food grain recognition and classification accuracies are found to be 97 percent and 85 percent, respectively, for all five types of food grain samples. The increased presence of foreign bodies in food grain samples drastically reduces recognition and classification accuracies. According to the study, a percentage of 10% foreign bodies mixed with food grain samples reduces its recognition percentage to as low as 60%.

Mesfin Fekadu et al. [10] used digital image processing techniques and an artificial neural network classifier to develop a system capable of assessing the quality of White pea bean sample constituents based on the Ethiopian standard for white pea bean. To remove the false regions, a preprocessing technique was used. This study employs various segmentation techniques to distinguish white pea bean sample constituents from one another and the background. This component includes thresholding, and the thresholding subcomponent extracts information from each of the three binary images to form an intermediate image known as the reconstructed image. The reconstructed image and the color structure tensor segmented image were merged to form the merged image, which contains complete information about the location of pest damage, discoloration, and rottenness in white pea bean kernels. To model white pea bean sample constituents, a total of 24 features (14 colors, 8 shapes, and 2 sizes) have been identified. A feed-

forward artificial neural network classifier with a back propagation learning algorithm, 24 input, and 7 output nodes, corresponding to the number of features and classes, was designed for the classification of White pea bean samples. The network is trained, and its performance is empirically and empirically based on supporting facts from the literature compared to other classifiers. The classifier's overall classification accuracy was 96.8 percent. Foreign, rotten and diseased, healthy, broken, discolored, shriveled, and pest-damaged kernel detection rates are 94.9 percent, 96.5 percent, 96.3 percent, 97 percent, 97.9 percent, 97 percent, and 97.6 percent, respectively. However, the white pea bean sample taken during image acquisition was exposed to light; as a result, the difference in illumination in a single image affects the values of the color features to be extracted and introduces shadows and shadings.

Mebatsion et al. [47] suggested a method for the classification of cereal grains, namely; barley, rye, oats, and wheat (Canada Western Amber Durum (CWAD) and Canada Western Red Spring (CWRS)). This was performed using morphological and color features. The combined model defined by morphological and color features achieved a classification accuracy of 98.5% for barley, 99.97% for CWRS, 99.93% for oat, and 100% for rye and CWAD, Nevertheless, the separation accuracy of the model for barley kernels needs improvement to achieve perfect classification.

Kruzilicova et al [48] attempted to use a Neural Network classifier to classify olive oil seeds. To characterize the features in the sample image, the system employs five different types of olive oil. The chemometric procedure was used, as well as absorbance at pre-selected optimal wavelengths. To select wavelengths in the olive oil type, a stepwise selection procedure in the linear discriminant analysis was used. The K-th nearest neighbor technique was the best for oil classification by variety, with 98.7 percent of the samples correctly classified; linear discriminant analysis was the best for oil classification by sensorial quality, with 89.0 percent of the samples correctly classified. The absence of some critical graphical output was a disadvantage of the KNN and LR techniques.

Pazoki and Pazoki [49] proposed a method for developing a digital imaging system and ANN capable of measuring geometric and shape-related parameters for distinguishing rain-fed wheat grain cultivars. Wheat was modeled using 6 colors, 11 morphological features, and 4 shape features in this work. This study, like most others, used ANN to classify wheat into six cultivars: Sardari, Sardari39, Zardak, Azar 2, ABR1, and Ohadi. The authors claimed that 86.48 percent classification accuracy was obtained. Many features were highly correlated with others, and if one was chosen, the rest would not contribute significantly to the classification model.

Habtamu Minassie et al. [51] proposed a method for categorizing Ethiopian coffee based on its growing region. This work is based on healthy coffee and aims to differentiate between different types of Ethiopian coffee using image processing technology. In this study, morphological and color features were extracted from coffee bean images from six Ethiopian regions: Bale, Harar, Jima, Limu, Sidamo, and Welega. Using Nave Bayes and neural network classifiers, the researchers assessed the classification accuracy of each selected feature set. The experiment was carried out using three different scenarios from the features data set: morphology, color, and both

morphology and color features. When morphology and color features were combined, neural networks produced the highest classification accuracy. Using neural network classification with morphological and color features parameters, the six regions of Bale, Harar, Jimma, Limu, Sidamo, and Welega were identified with a classification accuracy of 80.7 percent, 72.6 percent, 56.8 percent, 96.8 percent, 95.4 percent, and 61.9 percent, respectively. The overall performance, in this case, was 77.4 percent. Because the classification accuracy was obtained below laboratory settings, it had some limitations. The quality of the camera, the image acquisition environment, and other imaging factors can all have an impact on the outcome.

Tom Patten et al. [52] proposed an image-based system for the automatic analysis of *Heliothiszea* insects. This insect eats corn crops, and various insecticidal bio-toxins have been developed to kill it or stop its growth. The insect was analyzed in the proposed system using three steps: insect segmentation, regional processing, and instar and life classification. The proposed algorithm employs a back-propagation neural network and yields average values of 84 percent, 95 percent, and 66 percent for insect count, instar, and life, respectively. Our system was created and built specifically for *Heliothiszea* insect images. It could, however, be easily extended to handle other types of insect images, such as (Cabbage looper) images.

Table 2-5-1: Summary of related works

Author	Topic	Purpose/Objective	Methodology	Findings	Limitations
Zayas I., Pomeranz Y., and F. S. Lai [43]	Discriminate between wheat & non-wheat components in a grain sample	The use of image analysis to discriminate between wheat and non-wheat components in a grain sample	Two methods presented, multivariate discriminate and a structural prototype method for pattern recognition	Accuracy of quality classification is 90% and the accuracy of grading is 92%	The method requires to manually orient the kernels & misclassification of irregularly shaped stones as wheat
Lai FS, Zayas I, Pomeranz Y [44]	Application of pattern recognition techniques in the analysis of cereal grains	Pattern recognition techniques for identifying and classifying cereal grains	presented two methods, multivariate discriminate and a structural prototype method for pattern recognition	100% accurate prediction for the samples used in the study	Selection of variables to create a reliable pattern to characterize & distinguish specific grain from other.
N. S. Visenl, J. Paliwall [45]	Image analysis of bulk grain samples using neural networks	Proposed algorithms to acquire and process color images of bulk grain samples of five grain types	Back propagation neural network based classifier was developed	Classification accuracies of over 98% were obtained for all grain types except oats	The accuracy level obtained for oats significantly lower than the other grain types
B. S. Anami, D. G. Savakar, Aziz Makandar, and P.H. Unki [46]	A neural network model for classification of bulk grain samples based on color and texture	Developed a Neural network approach to classify single grain kernel of different grains	Neural network classification algorithm is used	Grain recognition and classification accuracies are found to be 97% and 85%	Increased presence of foreign in grain samples reduces the recognition and classification accuracies drastically
Mesfin Fekadu [10]	Development of an automated grading	Develop an automated grading system of white pea bean based	Image processing techniques for	Classifier achieved an	During image acquisition, the white pea bean sample taken

	system of white pea beans using image processing techniques convergence with ANN	on morphological and color features using ANN	feature extraction, ANN, Naïve bayes used as classifier	overall classification accuracy of 96.8%.	was exposed to light, this causes difference in illumination in a single image affects the values of the color features
D.Kruzlicova, J.Mocak, E.Katsoyannos & E.Lankmayr [48]	Classification and characterization of Olive Oils Using Neural Network	Attempted to develop classification of olive oil seeds using Neural Network classifier	Chemo metrical procedure and absorbance at pre-selected optimal wavelengths were used	89.0% of the samples were correctly classified	Disadvantage of the KNN and LR techniques was the absence of some essential graphical output.
Pazoki and Pazoki [49]	Classification System for Rain Fed Wheat Grain Cultivars Using Artificial Neural Network	Proposed a method for the development of a digital imaging system using ANN extracting morphological features to differentiate b/w rains fed wheat grain cultivars	This work used ANN to classify wheat into 6 cultivars	The Authors claimed that 86.48% of classification accuracy was achieved	Do not address quality factors
Habtamu Minassie [51]	Image Analysis for Ethiopian Coffee Classification	Design an appropriate classification model of Ethiopian coffee varieties with respect to their growing region	Morphological and color features were extracted, Naïve Bayes and neural network classifiers	Overall classification accuracy obtained is 77.4%	The classification accuracy acquired is below laboratory settings
Tom Patten, Wenjing Li, George Bebis & Muhammad Hussain [52]	Automatic analysis of Heliothiszea insect by using of image analysis	Proposed a system for the automatic analysis of Heliothiszea insect by using of image analysis	Follows a back-propagation neural network	Result for insect count, instar and life with average value of 84%, 95% and 66% respectively	Implemented only for Heliothiszea insect images, it could be extended to Spodopteraexigua (Beet armyworm) and Trichoplusiani

2.5.3 Research Gap

We looked at previous research on the automatic classification and grading of agricultural products using digital image processing techniques. Various strategies for automating agricultural product classification and grading have been proposed over the years, it was noted. According to the literature, the proposed approaches vary depending on the type of agricultural commodity. This is due to changes in the types of symptoms detected in each product, as well as differences in the environmental conditions in which the product grows. As a result, a system designed for categorization and grading of any agricultural commodity could not be extended immediately to other goods. As a result, we present a model for automatic categorization and grading of Ethiopian red kidney bean in this thesis (RKB).

CHAPTER THREE

Methods

3.1 Overview

To meet the research aims and have appropriate information on the study, the literature on the current development of image analysis related to cereals quality detection has been researched. Among other things, the review reviewed books, previous study endeavors, the Internet, and articles. These insight reviews were utilized to select image analysis techniques and tools that have been applied on agricultural product quality identification and disease detection that are relevant to our work. This chapter presents the proposed architecture and methodologies for developing a prototype for RKB classification and grading.

3.2 The Proposed Architecture

Figure 3-2-1 depicts the suggested architecture. The five components of the proposed system architecture for analyzing RKB data are preprocessing, segmentation, feature extraction, classification, and grading. The preprocessing component removes erroneous sections based on their size. Individual RKB sample elements are separated from the background and from each other in the segmentation component using a combination of color structure tensor and thresholding. The feature extraction component computes representative color, shape, and size features. Finally, the classification component employs artificial neural networks (ANN) to categorize RKB sample pieces into one of five groups.

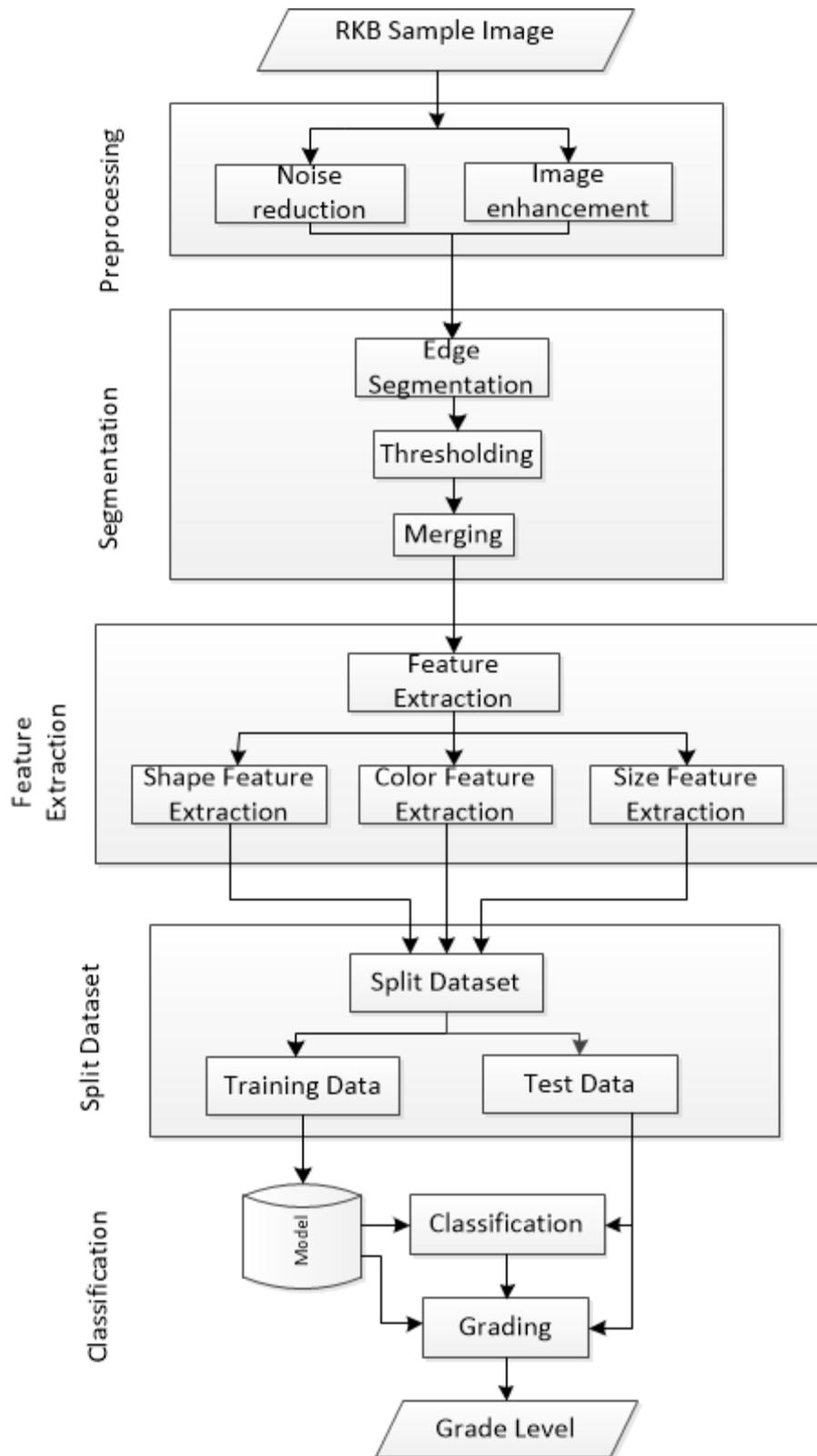


Figure 3-2-1: The system architecture for the proposed system

3.3 RKB Image Data Collection

Sampling is one of the most significant procedures in RKB bean classification and quality assessment. According to current manual system practice, the sample drawer selects a representative sample of 3 kg per 10 tons of a vehicle, which is the average carrying capacity of a truck, upon arrival. For green analysis, just 125g of the 3kg is used. The remainder was divided into two halves. The first half is given to the customer, while the second half is kept in the grading department for a month. We'll take RKB bean samples from these sampled data because we require certified RKB for our research. As a result, we collected five independent 100 gram samples of RKB from each RKB manufacturing zone, all of which came from the same plant but were delivered to the quality inspection facility at various times. As a result, a section of the 500 gram RKB bean image was taken from a certain place.

Images of RKB were taken with a Huawei TRT-L21A 12Mega Pixel digital camera. When the photographs were taken, the camera was placed on a stand that allowed for easy vertical movement while still providing solid stability. The camera was situated 130 mm away from the sample table to obtain crisp photos of RKBs. During the photographing process, samples were placed on a white background table. The RKBs were strewn across the table, none of them touching. The distinction between RKBs was kept to make image segmentation easier. To create consistent or balanced lighting, a 100W incandescent lamp with a rated voltage of 220V was used in all of the testing. Before taking any photographs, the lighting system was turned on for about 5 minutes to help with image stabilization. We took the samples in a controlled environment to reduce the impact of ambient light. Images with a resolution of 2818 x 1826 pixels were taken.

3.4 Image Pre-Processing

Figure 3-4-1 depicts the five components utilized to prepare the input image for analysis and classification. The component is responsible for preparing the input image for segmentation. Because the background of the acquired pictures is not smooth, some groups of pixels in a segmented image may have the foreground grey level value while being surrounded by the background. These locations are naturally in the background, however they appear to be foreground elements. In this paper, these are referred to as fake areas, and they are removed during image pre-processing. To do so, we must restore their grey level value to that of the background. We can determine whether something is true or false by computing the size of a region. We looked at foreground objects to see if they were false or not, and determined that false regions are defined as clusters of pixels smaller than 500 pixels in size. The saturation and hue photographs serve as the foundation for the pre-processing method. To identify false zones, the area of each zone is computed, and the pixels in those regions' values are set to the background grey value.

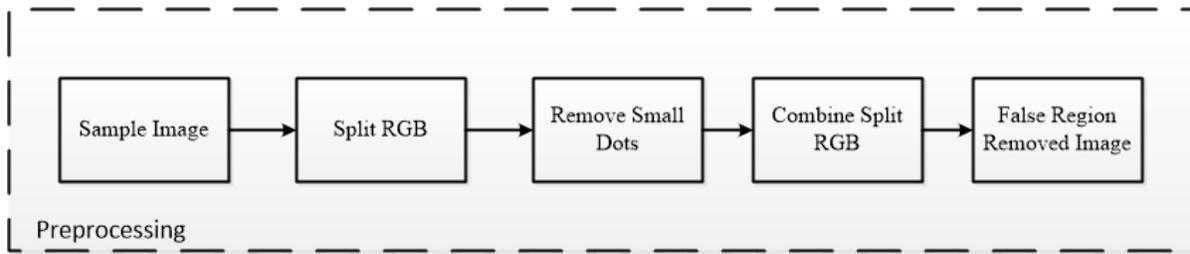


Figure 3-4-1: The pre-processing flowchart

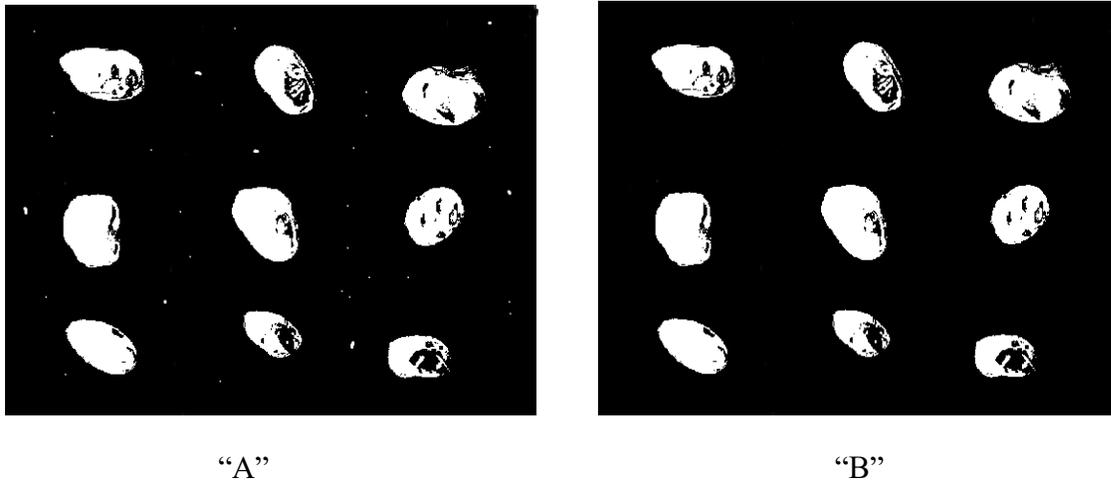


Figure 3-4-2: “A” False Regions Shown as Tiny White Spots in the Background, “B” RKB Kernels after False Region Removal

3.5 Image Segmentation

The segmentation component of our suggested architecture is in charge of distinguishing RKB sample parts from each other and from the image's backdrop. This component includes thresholding, and the thresholding subcomponent collects information from each of the three binary images to create an intermediate image known as the reconstructed image. The merging sub-component completes the segmentation component's last task. The merging sub-component combines the outputs of the structure tensor segmentation and thresholding sub-components to produce a new binary picture known as a merged image. A merged image contains all of the information needed to extract the classification features. The proposed segmentation algorithm is thresholding algorithms. It is based on thresholding segmentation approaches that produce results. Figure 3-5-1 depicts the planned flowchart segmentation.

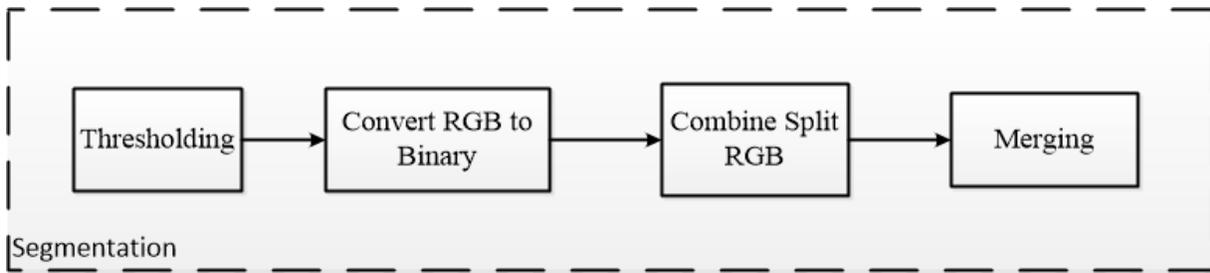


Figure 3-5-1: The proposed flowchart for image segmentation

3.5.1 Segmentation Using Thresholding

Thresholding is one of the segmentation procedures available, as discussed in Chapter Two. It is operated using greyscale visuals. Thresholding produces a binary image in black and white. Successful image segmentation, on the other hand, results in a complete separation of the image's backdrop and foreground with no information loss. Because the color, shape, and size information of each grain is lost to the backdrop, this information loss has a significant impact on the RKB grain recognition process.

Algorithm 3-5 shows the step-by-step instructions for separating red kidney beans for classification and grading using segmentation-based thresholding.

```

Input: - The preprocessed image P, and the Preprocessed Saturation image bS
Output: - A segmented image ready for feature extraction
Convert the RGB components into binary images bR, bG and bB
Segment P using Color Structure Tensor Segmentation Algorithm to get T
For each pixel i in P
  If (bR(i)=1 and bG(i) ==0 )
    bS(i)=0;
  End If
  If (bR(i)=1 and bB(i) ==0 )
    bS(i)=0;
  End If
End For
For each pixel i in bS
  If (bS(i) >=1 )
    T(i)= bS(i);
  Else If (bS(i)==0 and T(i) >0)
    T(i)= 255;
  Else
    T(i)= 0;
  End If
End For
Label T;
Make a merged image with size equal to the size of bS and having all the
pixel values equal to zero, Z
For each pixel i in bS
  If (T(i) ==0 )
    Z(i) = 0;
  Else If (bS(i)==0 and T(i) > 0)
    Z(i)= 255;
  End If
End For
Return Z;

```

Algorithm 3-5: The Proposed Segmentation Algorithm

Figure 3-5-2 below depicts the result of the proposed algorithm used for segmenting and identifying each red kidney bean image



“A”

“B”

Figure 3-5-2: “A” Original image, “B” Result of Thresholding

3.5.2. Merging Segmented Images

Each of the three greyscale component images in an RGB image contains different information. Some of the information in the RGB image may be present in one of the component images but not in the others. Based on empirical research, we discovered that distinct area information for discoloration is present in the blue and green component binary images, but not in the red component binary image. As a result, the data from RGB component images must be combined to produce a reconstructed image. The RKB kernels' boundary information, on the other hand, is missing from this rebuilt image. As a result, the reconstructed image must include kernel boundary information. This information is always present in the segmented color structure tensor image. Figure 3-5-4 depicts the reconstructed image of the images depicted in Figure 3-5-3 above.

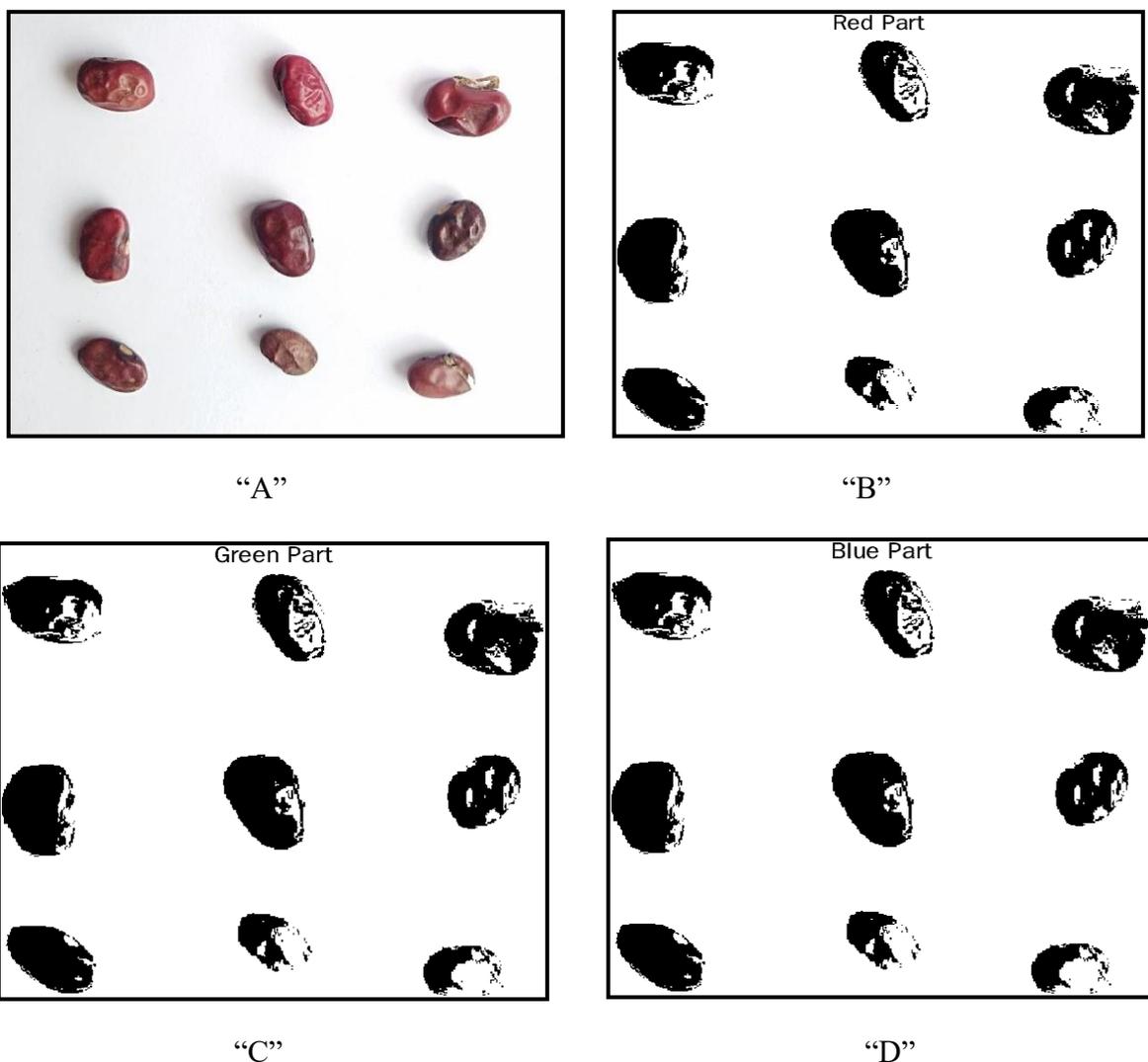


Figure 3-5-3: RKB Image and its RGB Components “A” Original RGB Image “B” Binary Image of the Red Component “C” Binary Image of the Green Component “D” Binary Image of the Blue Component

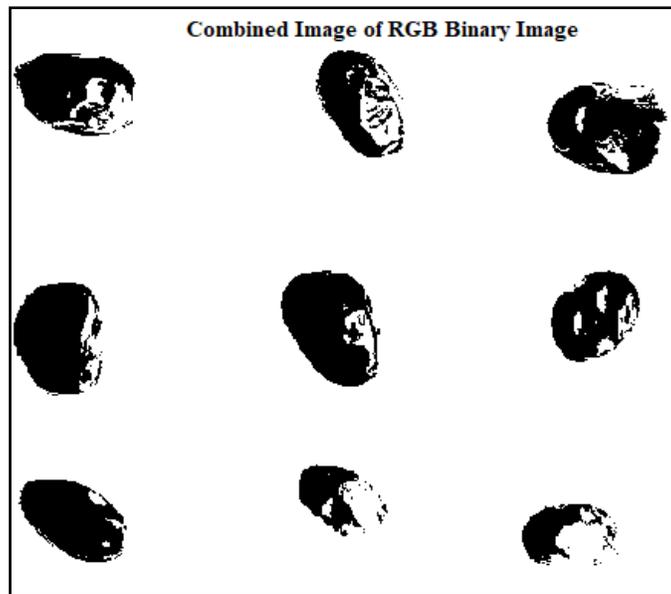


Figure 3-5-4: Reconstructed images of the Red, the Green and the Blue Binary Images

3.6 Feature Extraction

The descriptive features of the RKB sample parts are extracted by the feature extraction component. This component is subdivided into three parts: color feature extraction, shape feature extraction, and size feature extraction. One of the goals of this work, as stated in Chapter One, is to use digital image processing to identify five categories of RKB sample elements based on their color images (DIP). The examination of quantitative data derived from images is required for this work. Processing all quantitative data from the RKB sample, on the other hand, is computationally inefficient. As a result, we select and extract RKB sample constituent features that are representative of the RKB sample. Color, size, and shape are the three types of attributes that can be extracted from RKB images to characterize the elements of the RKB sample constituents; we discovered 14 color features, 2 size features, and 8 shape features.

3.6.1 Color Feature Extraction

Color is one of RKB kernels' visual characteristics. It aids in distinguishing between different types of RKB samples. For example, the color features of contrast and defect RKB kernels can be used to distinguish them satisfactorily. Furthermore, color has the ability to separate foreign matters from RKB kernels.

Fourteen color features were chosen to represent the color characteristics of RKB sample constituents. The first set of extracted color features is based on the RGB color model. These characteristics are the mean values of each image's red, green, and blue components as computed from its three-color channel functions using Equation (16). The channels of the RGB color model are represented by the pixel value functions $r(x, y)$, $g(x, y)$, and $b(x, y)$ in this equation [38].

$$R = \frac{1}{N} \sum_{k=1}^N r(x, y) \quad (3.1)$$

$$G = \frac{1}{N} \sum_{k=1}^N g(x, y) \quad (3.2)$$

$$B = \frac{1}{N} \sum_{k=1}^N b(x, y) \quad (3.3)$$

Where x , y , k , and N are positive integers.

In addition to these, we identified three additional color features based on the RGB model that correspond to the damage areas within an RKB kernel: spot-red, spot-green, and spot-blue. Based on the area of the damaged region within an RKB kernel, these features are calculated using Equation (3.1). These characteristics are discussed in Chapter Four. Table 3-6-1 shows a screenshot of the numerical values of the 14 color features corresponding to 12 healthy RKB kernels. The columns titled SpotR, SpotG, and SpotB in this table represent the spot-red, spot-green, and spot-blue features, respectively.

	Red	Green	Blue	AverageRGB	Hue	Saturation	Intensity	AverageHSV	SpotH	SpotS	SpotV	SpotR	SpotG	SpotB
1	10360	0.7847	0.9439	0.9044	2.6331	0.3327	0.1565	0.7955	1.2847	0.5777	0.4466	0.6811	0.0317	0.5701
2	10207	0.7878	0.9658	0.8949	2.6485	0.3550	0.1458	0.8127	1.3135	0.5941	0.4772	0.6879	0.0023	0.5058
3	9586	0.7789	0.8834	0.8873	2.5497	0.3045	0.2140	0.7720	1.2905	0.4902	0.3981	0.6398	0.2093	0.5970
4	10548	0.7651	0.8759	0.8545	2.4954	0.2721	0.2229	0.7045	1.1995	0.4963	0.4108	0.6233	0.1894	0.4979
5	8345	0.7643	0.9225	0.8090	2.4957	0.3120	0.1779	0.7507	1.2406	0.5318	0.4399	0.6157	0.1207	0.2874
6	12394	0.8084	0.9399	0.8616	2.6099	0.2832	0.1730	0.7739	1.2301	0.5842	0.4482	0.6181	0.0238	0.2951
7	8684	0.7645	0.9119	0.8688	2.5453	0.3063	0.1827	0.8103	1.2992	0.5509	0.4271	0.6493	0.0935	0.4951
8	10634	0.7910	0.9480	0.8710	2.6100	0.2699	0.1629	0.7938	1.2265	0.5766	0.4412	0.6384	0.0190	0.3943
9	10609	0.7887	0.9541	0.8325	2.5753	0.3066	0.1474	0.7691	1.2231	0.5885	0.4584	0.6210	0.0110	0.2161
10	11336	0.7843	0.8543	0.8117	2.4502	0.2544	0.2213	0.7119	1.1876	0.4896	0.3873	0.5721	0.2146	0.3142
11	10063	0.7854	0.9361	0.8145	2.5359	0.2791	0.1894	0.7744	1.2429	0.5883	0.4408	0.5967	0.0146	0.1483
12	9344	0.7690	0.9576	0.8191	2.5458	0.2776	0.1913	0.7699	1.2388	0.5849	0.4566	0.6397	0.0164	0.2297

Figure 3-6-1: Screenshot Showing the Values of 14 Color Features for 12 Healthy RKB Kernels

The second set of color features measured in this work is based on the HSV color model. As described in Chapter Two, the HSV color model is the other most commonly used color model. In this model, color is described by three components: hue, saturation, and value (intensity). The colors of the RGB space are usually not easy for humans to interpret. However, the hue, saturation, and value space, HSV color space is, by contrast, intuitive. Hue is an attribute associated with the dominant pure color such as pure blue, pure red, etc. Saturation is the amount of white light that is mixed with a hue while intensity (value) is defined as a measure of the brightness of light.

In the HSV color space, the intensity attribute is decoupled from the color information. Moreover, the hue (H) and saturation (S) attributes are closely related to the way human beings perceive color. The H, S, and V attributes can be derived from the RGB model components as discussed in Chapter Two. In this research work, the color features are extracted using the mean values of each component of the HSI model and are calculated for each foreground region by using Equation (3.1), (3.2), and (3.3). In this equation, the functions $h(x, y)$, $s(x,y)$, and $v(x, y)$ are the respective channels of the HSV color model [38].

$$H = \frac{1}{N} \sum_{k=1}^N h(x, y) \quad (3.4)$$

$$S = \frac{1}{N} \sum_{k=1}^N s(x, y) \quad (3.5)$$

$$V = \frac{1}{N} \sum_{k=1}^N v(x, y) \quad (3.6)$$

Where x , y , k , and N are positive integers.

In addition, based on the RGB model that corresponds to the damage areas within an RKB kernel, we identified three additional color features: spot-hue, spot-saturation, and spot-value. Based on the area of the damaged region within an RKB kernel, these features are calculated using Equations (3.1), (3.2), and (3.3). These six distinguishing characteristics of RKB kernels are extremely useful in distinguishing between healthy and defective kernels. Table 3-6-1 shows the values of these features for 10 healthy and 10 defect kernels.

Table 3-6-1: Color Feature Values for Samples of Healthy and Defect Kernels

Healthy						Defect					
Spot-Hue	Spot-Saturation	Spot-Value	Spot-Red	Spot-Green	Spot-Blue	Spot-Hue	Spot-Saturation	Spot-Value	Spot-Red	Spot-Green	Spot-Blue
1.28	0.58	0.45	0.68	0.03	0.57	1.15	0.37	0.31	0.61	0.38	0.64
1.31	0.59	0.48	0.69	0.00	0.51	1.18	0.38	0.36	0.62	0.37	0.75
1.29	0.49	0.40	0.64	0.21	0.60	1.18	0.32	0.34	0.60	0.38	0.59
1.20	0.50	0.41	0.62	0.19	0.50	1.14	0.31	0.32	0.59	0.47	0.60
1.24	0.53	0.44	0.62	0.12	0.29	1.15	0.40	0.34	0.63	0.35	0.73
1.23	0.58	0.45	0.62	0.02	0.30	1.20	0.51	0.37	0.65	0.09	0.75
1.30	0.55	0.43	0.65	0.09	0.50	1.19	0.44	0.36	0.61	0.26	0.62
1.23	0.58	0.44	0.64	0.02	0.39	1.16	0.35	0.29	0.61	0.43	0.75
1.22	0.59	0.46	0.62	0.01	0.22	1.14	0.30	0.36	0.58	0.41	0.48
1.19	0.49	0.39	0.57	0.21	0.31	1.15	0.44	0.33	0.61	0.16	0.66

The spot-hue, spot-saturation, and spot-value, features are calculated using Equations (3.4), (3.5), and (3.6) respectively. These features are discussed in Chapter Four. The results of the numerical values of all the color features corresponding to 12 healthy RKB kernels are shown in Table 3-6-1. In this table, the columns titled SpotH, SpotS, and SpotV represent spot-hue, spot-saturation, and spot-value features respectively.

3.6.2 Size Feature Extraction

The size of RKB sample constituents is determined using two size features, namely area, and perimeter features, in this study. The total amount of pixels corresponding to a single kernel, as mentioned in Chapter Two, is called an area. Similarly, the perimeter of a kernel region is the total number of pixels around it. Area, in addition to being a single feature, is used to derive all of the color features, as illustrated in Equations (3.1) and (3.2). The total of each color feature's color values is divided by the area to calculate the value of each color feature. Figure 3-6-2 depicts the proposed approach for extracting size features.

	Area	Perimeter
1	0.4786	134
2	0.4886	179
3	0.4019	132
4	0.3227	120
5	0.4891	150
6	0.6843	153
7	0.5870	133
8	0.3832	133
9	0.3465	164
10	0.5556	115
11	0.5625	143
12	0.4516	105

Figure 3-6-2: Screenshot of the Values of 2 Size Features for 12 Discolored RKB Kernels

3.6.3 Shape Feature Extraction

As described in Chapter Two, shape descriptors are numbers that are computed from a two-dimensional shape. The shape descriptors can thus be considered as an approximate description of the shape. Shapes can be examined for their similarity by way of using their respective shape descriptors. Hence, shape similarity somehow corresponds to the similarity of the shape descriptors.

The shape of the RKB kernel is believed to be good for the discrimination between foreign matters and RKBs. Moreover, it is also a helpful tool in the separation process of broken and whole RKB kernels. The eight shape descriptors identified in this research work are a count of convex hull sides, aspect ratio, ovality, triangularity, convexity, solidity, major axis to area ratio, and Fourier descriptor. The count of convex hull polygon sides is the number of sides of convex hull polygon. Triangularity is defined as the area of a triangle with a base equal to the minor axis of the kernel and height equal to the major axis. The major axis to area ratio is the ratio of the major axis to the area of the RKB kernel. The rest of the shape descriptors are explained in Chapter Two. Sample data of these 8 shape features is shown in Table 3.4. The eighth shape descriptor is the Fourier Descriptor (FD), computed as follows every connected object has a closed contour that can be represented as a sequence of the pixel coordinates $x(t)$, $y(t)$ where $t = 0, \dots, N - 1$. The coordinates can be considered to be sampling values[55].

$$(x(t), y(t)) = f\left(\frac{2\pi}{N}t\right)$$

of a continuous, closed curve $f : [0, 2\pi] \rightarrow \mathbb{R}^2$ such that f can be extended continuously to a 2π -periodic function. When the function f (or, more generally, any signature function $g(x(t), y(t))$ derived from the coordinates) is expanded into a Fourier series, a fixed number of discrete Fourier coefficients approximately represents the contour shape. This allows for data reduction.

We used FDs to describe the contour of an object. First, we compute a set of FDs for a healthy RKB kernel. Then, we compute the FDs of an unknown object and compare it to the known RKB kernel by ignoring the first component of the descriptors. The known object, whose FDs are the most similar to the unknown object's FDs, is the object the unknown object is classified to.

	ConvexHull	AxisRatio	Ovality	Triangularity	Convexity	Solidity	MajorAxis/area	Fourier
1	0.8340	375.2500	0.9925	0.6415	0.0216	0.9788	0.0122	0.0906
2	0.7474	370.3710	0.9946	0.6401	0.0169	0.9834	0.0130	0.0900
3	0.7109	364.9950	0.9776	0.6512	0.0148	0.9854	0.0138	0.0534
4	0.6748	401.6500	0.9654	0.6594	0.0316	0.9694	0.0136	0.1316
5	0.6592	341.9450	0.9816	0.6486	0.0138	0.9864	0.0154	0.0707
6	0.8666	401.8060	0.9947	0.6400	0.0138	0.9864	0.0109	0.1162
7	0.7311	346.0320	0.9882	0.6442	0.0172	0.9831	0.0142	0.1030
8	0.8430	376.0280	0.9904	0.6428	0.0213	0.9792	0.0120	0.1040
9	0.7987	377.2020	0.9838	0.6471	0.0145	0.9857	0.0124	0.0190
10	0.7939	408.6540	0.9696	0.6566	0.0339	0.9672	0.0121	0.0948
11	0.8964	359.9270	0.9975	0.6382	0.0150	0.9852	0.0119	0.0835
12	0.6746	363.8130	0.9827	0.6478	0.0199	0.9805	0.0143	0.0375

Figure 3-6-3: Screenshot of the Values of 8 Shape Features for 12 Healthy RKB Kernels

Fourier descriptors are invariant to scaling, translation, and rotation. These properties make Fourier descriptors suitable to compare object shapes having a range of different sizes and orientations. Sample data showing Fourier descriptors of 12 healthy RKB kernels is shown in Figure 3-6-3.

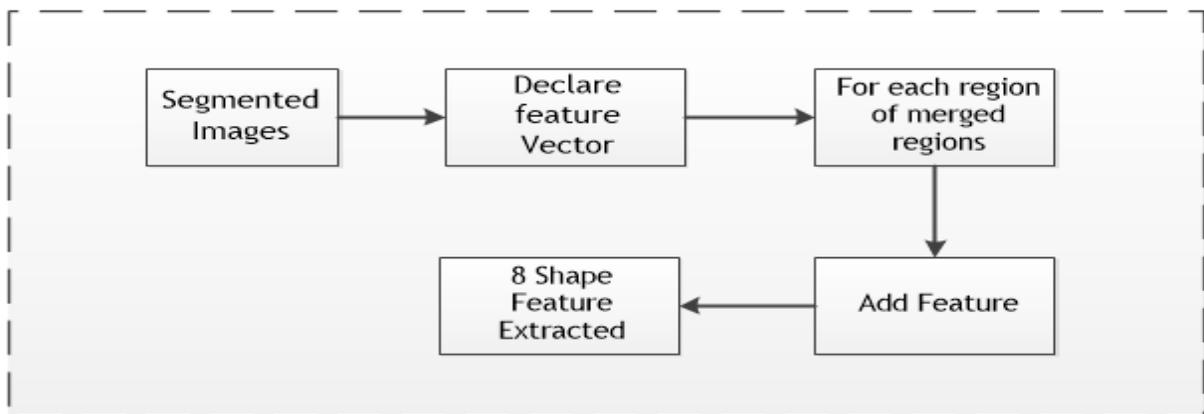


Figure 3-6-4: Shape Features Extraction Algorithm

3.7 Classification

The classification component contains the ANN classifier and the class count sub-components. The ANN classifies RKB samples into five classes. The class count subcomponent is responsible for counting the number of sample constituents belonging to each class. Although there are other methods like mathematical functions, rule-based algorithms or statistical methods available for classification, we chose ANNs over others. There are several reasons for choosing neural

networks over other methods for this research work. The classification of grain kernels cannot be easy using unique mathematical functions. This is due to the variation in morphology, color, and texture of the grain kernels under consideration. ANNs have the potential of solving problems in which some inputs and corresponding output values are known, but the relationship between the inputs and outputs is difficult to translate into a mathematical function.

When compared to other methods, ANNs can tolerate noise better and exhibit low classification error rates [59]. Moreover, compared to statistical methods, ANNs using the BP network could be easily modified to accommodate more features. To add empirical experience to the above claims, we trained the naïve bayesian classifier and ANN classifier on the same training data set. Naive Bayes is a kind of classifier that works on the Bayes theorem. Prediction of membership probabilities is made for every class such as the probability of data points associated with a particular class. Naive Bayes classifiers conclude that all the variables or features are not related to each other. The Existence or absence of a variable does not impact the existence or absence of any other variable.

The neural network architecture in this work is a three-layered F-F network with sigmoid hidden and softmax output neurons. Such a network can classify vectors arbitrarily well, given enough neurons in its hidden layer. The input layer contains 24 neurons corresponding to every 24 inputs and the output layer consists of 5 neurons corresponding to each 5 output class. Softmax is a neural transfer function. Transfer functions calculate a layer's output from its net input. The network is designed to have only one hidden layer consisting of 45 nodes. The hidden layer of the neural network is composed of 45 neurons. This number of neurons in the hidden layer is selected empirically based on the performance it exhibited over a smaller and larger numbers of neurons. Moreover, the decision to use only 1 hidden layer is made based on facts found in the literature. There is no reason to use any more than one hidden layer. The network is designed to use B-P algorithm training. To measure the performance of the network during the training phase, we preferred to use the cross-entropy error function over the mean square error (MSE). Compared to MSE, the cross-entropy function is proven to accelerate the backpropagation algorithm and to provide good overall network performance. The architectural design of this ANN is depicted in Figure 3-7-1.

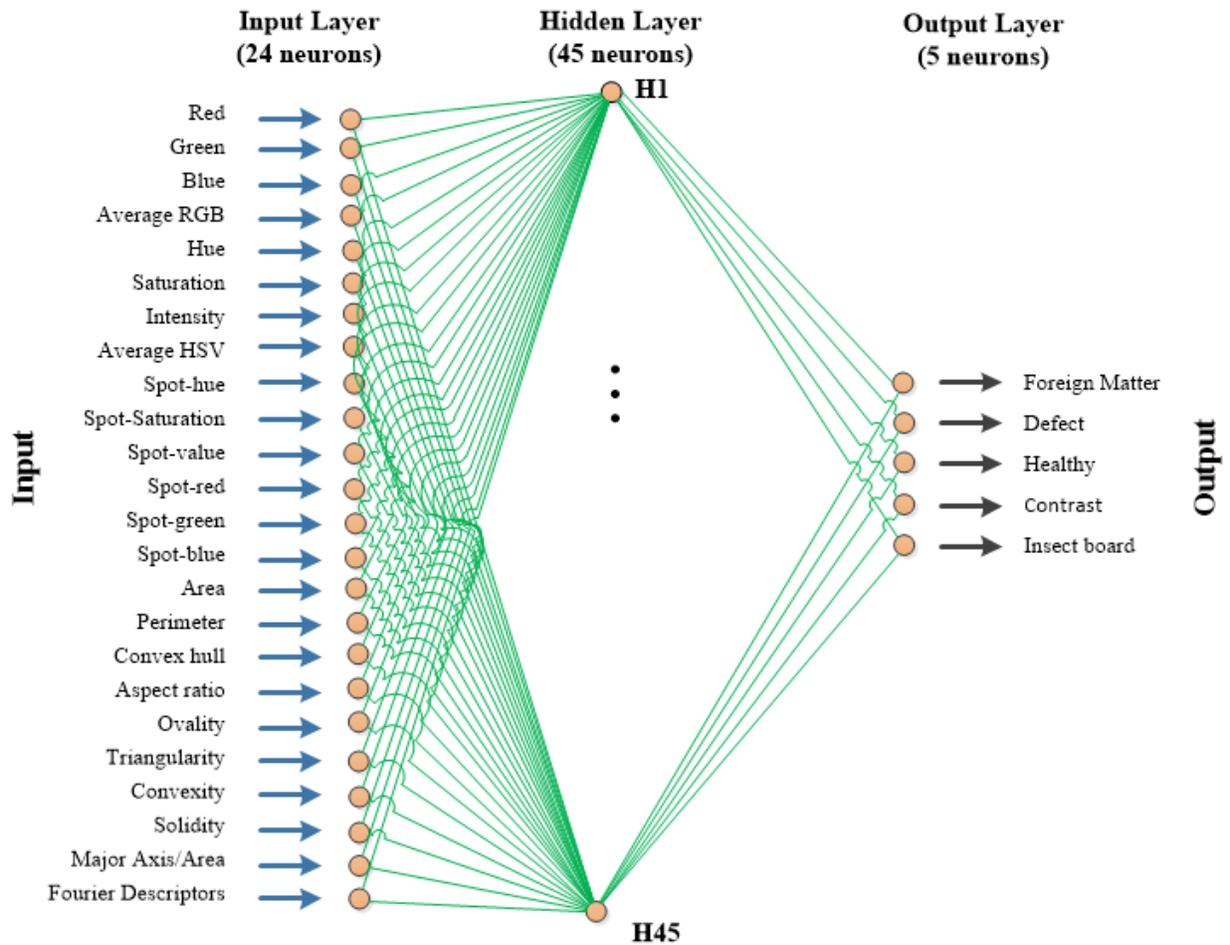


Figure 3-7-1: Design of ANN Used for the Classification of RKB Sample

3.7.1 Naïve Bayes Classifier

A Bayesian classifier, as defined in section 2.4.4, is a statistical classifier based on probability distribution. Based on the observable features, it classifies an object into the class to which it is most likely to belong. Naive Bayes is a classifier that employs the Bayes Theorem. It forecasts membership probabilities for each class, such as the likelihood that a given record or data point belongs to a specific class. The most likely class is the one with the highest likelihood. This is also referred to as Maximum A Posteriori (MAP). The Naive Bayes classifier presumes that all features are unrelated to one another. The presence or absence of a feature does not influence the presence or absence of any other feature. This classifier is simple to implement and computationally fast and performs well on large datasets having high dimensionality.

On the RKB samples, the Nave Bayes classifier is used. The prediction output is generated by the classification findings. Nave Bayes is a subset of Bayesian decision theory that has recently gained interest. This method is used because it requires little storage and has a quick training procedure. The goal of this method is to devise a rule for allocating future items from a set of objects to a class given that the vectors of variables identify the future objects. There is a

supervised classification issue found very commonly and several methods have been designed to construct such rules. Any complicated repetitive parameter estimation mechanism is not required in this algorithm which means that huge data sets can apply this algorithm. Even the unskilled users using the classifier technique can understand the classification process since it is easy to be interpreted.

3.8 Grading of RKB

The grading of RKB is done based on the agricultural procedure of Ethiopia Commodity Exchange standards. According to the standard RKB is divided into five classes (Grade 1 up to Grade 5) and the six ones low Grade (LG).

The proposed grading method is based on the ECX's grading rules. After the image has been segmented, the feature extraction process begins. Size (e.g., area) and shape (e.g., eccentricity, centroid, minor and major axes) are the two attributes that provide efficient information to distinguish RKB from foreign matter based on inspection of the RKB grain. To distinguish RKBs from foreign particles, the system or grading algorithm is programmed to learn the area of RKBs.

Once the system has become accustomed to the RKB area, it is employed as a rudimentary method of recognizing the RKBs. For example, in our scenario, the area of the RKB is between 0.32 and 0.68, indicating that any object with an area outside of this range can be classified as a foreign particle. However, there are foreign particles in places that fall inside that range. As a result, in order to highlight the RKBs solely, other attributes, such as shape feature, have to be considered as well. We used the minor axis, major axis, and eccentricity as shape features for the desired purpose, among other types of shape features. We may be able to simply obtain the desired features by using a region props function. Most of the time, these characteristics are one-of-a-kind. This means that the RKB has a limited range of minor axis, major axis, and eccentric value values. The minor axis, major axis, and eccentricity of an extracted RKB, for example, are 6.65–7.00 mm, 4.70–6.13 mm, and 7.31–9.24 mm, respectively. At this stage, we have two separated zones, one with solely RKBs and the other with foreign particles. But to proceed from here, it was needed to come up with a different way of conducting the grading. This is why the idea of calibration is used. Calibration “Calibration in measurement technology is the comparison of measurement values delivered by a device under test with those of a calibration standard of known accuracy.” Thus, to calibrate the grading system, we used different sample images with already known grade levels. The grading system is performed using the following steps.

- The RKBs are separated from the foreign particles using size and shape features.
- The total area of the RKBs is computed.
- The total area of the foreign particles is computed.
- The ratio of the total area of the foreign particles to the total area of the RKBs is computed and for our purpose, we named it, “weight”.
- The weight value labeled is with the corresponding grading level

We identified the weight range values through empirical analysis. Based on the weight value, the sampled RKBs is grouped into one of the different grade levels

3.9 Evaluation Methods

A test dataset was used to evaluate each model, which was constructed using the training dataset. The classifiers' model performance was returned as an output, which included performance matrices and percentage accuracy metrics for each class, as well as a confusion matrix. A confusion matrix is a type of contingency table that is used to drive true positives, true negatives, false positives, and false negatives, showing the correct/incorrect class assignment of data.

Performance evaluation of classification model is important for understanding the quality of the model, refining the model, and choosing the adequate model. The confusion matrix displays the number of correct and incorrect predictions made by the model compared with the actual classifications in the test data [57]. The confusion matrix for a classifier with two classes true and false is presented in Table 3-9-1.

Table 3-9-1: The Confusion Matrix of a Classifier with Two Classes

Classes Predicted \ Current Classes	True Class	False Class
True Class	True Positive	False Positive
Flase Class	True Negative	False Negative

The number of correctly predicted values relative to the total number of predicted values is specified by a precision parameter that takes values between 0 and 1. Precision equal to 0 indicates that the model has no predictive power and is not conclusive [57] and precision equal to 1 indicates both predictive power & conclusive. Evaluation of the classification and grading algorithms is one of the key points in any digital image processing.

The performance evaluation metrics commonly used in analyzing the results of classification algorithms applied are accuracy measure and error measurement using relative error (mean percentage error) propagation [57].

The accuracy is the proportion of the total number of correct predictions and is calculated as the ratio between the number of cases correctly classified and the total number of cases [57].

$$\text{Accuracy (recognition rate)} = \frac{\text{Number of correctly predicted}}{\text{number of_total_samples}} = \frac{TP \text{ Rate} + FN \text{ Rate}}{N + P}$$

Where, **P**: positives which refer to the total number of positive tuples.

N: negatives which refer to the total number of negative tuples.

TP: True positives refer to positive tuples that were correctly labeled by the classifier.

TN: True negatives refer to negative tuples that were correctly labeled by the classifier.

FP: False positives refer to the negative tuples that were mislabeled as positive.

FN: False negatives refer to the positive tuples that were mislabeled as negative.

The Error indicates the proportion of cases classified incorrectly.

$$\text{Error (misclassification rate)} = 1 - \text{Accuracy}$$

Where accuracy is the proportion of the total number of correct predictions.

3.10 Summary

Preprocessing, segmentation, feature extraction, and classification are the four components of the system design we suggested. Noise and false regions are removed by the preprocessing component. The preprocessing component's output was sent into the segmentation component. Our innovative segmentation technique, which combines color structure and thresholding segmentation techniques, is included in the segmentation component. The third component, feature extraction, extracts features from the segmentation component's output. This component retrieves a total of 24 features (14 colors, 8 shapes, and 2 sizes) for the image. Modeling the various properties of RKB sample constituents was the goal. The classification component, the fourth component, categorizes RKB data using the characteristics retrieved by the feature extraction component. This component includes a neural network classifier with 24 input nodes and 5 output nodes, which correspond to the number of input features and output classifications, respectively. The following chapter contains experimental findings to recommend the optimum model for RKB classification and grading.

CHAPTER FOUR

Experimentation and Discussion

4.1 Overview

In this chapter, we report a set of experimental results carried out to test the effectiveness of our proposed system. Accordingly, the type of classification algorithms, the data set used, and the results achieved in the classification process have been discussed. Alongside these, the discriminative power of color, size, and shape are tested and compared. MATLAB version R2018a tool is used to develop the prototype of the system. Moreover, the specification of the computer on which the system is implemented is an Intel Core i7 Desktop computer with 8GB RAM and a 2.3 GHz processor.

4.2 Data Set

A total of 582 RKB kernels (small matrix used for blurring, sharpening, embossing, edge detection, and more.) and foreign matter are prepared to train, validate and test the proposed model. These 582 RKB sample constituents are separated into their corresponding 5 classes based on their characteristics. Therefore, we finally have 5 outputs each corresponding to each of the 5 classes. The data were partitioned randomly into training, validation, and test sets. Image acquisition is done using Huawei Model TRL with a specification of 12.1 megapixels. The images taken are all 24-bit color JPEG format.

The number of RKB kernels per image is different for the different classes as shown in Table 4.1 during image acquisition; the camera is mounted on a stand which provides easy vertical movement. The distance between the camera and the sample was fixed at 14 cm to maintain the same vertical distance on each image taken. During background color selection, we compared white, blue-black, and light blue colors. We observed that the white color makes a good contrast with the foreground objects and achieved better segmentation results. Consequently, for each image, a white background is used. The samples of RKB are placed directly under the camera for image acquisition.

For this research 70/30 is chosen, where 70% of the data is used training, the rest of the data is used for validation and testing each consisting of 15% of the input data. We select 70/30% because, as suggested in [58], for very large datasets, 80/20% to 90/10% can be used; however, for small dimensional datasets, either 60/40% to 70/30% can be used.

The training set is presented to the neural network and naïve Bayesian classification during training. The training set is used to fine-tune the weights of the network. Whereas, the validation set is used to measure the network's generalization ability, and to halt training when generalization stops improving. The testing data do not affect training and so provide an independent measure of network performance during and after training. As this is a supervised effort, the training data

needs to be labeled. The labels are presented to the neural network as binary code. Since there are 5 classes into which the RKB sample constituents are to be classified, the corresponding number of bits in the binary code is also set to five. These classes and the number of images used for each in the training process are shown in Table 4-2-1.

Table 4-2-1: Data set description

Target Class Description	Binary Code	Number of Samples
Foreign Object	00000001	40
Healthy	00000010	210
Defect	00000100	143
Insect Board	00001000	45
Contrast	00010000	144
TOTAL		582

4.3 Experimenting Segmentation

Threshold segmentation

Threshold segmentation is the simplest method of image segmentation and also one of the most common parallel segmentation methods. It is a common segmentation algorithm that directly divides the image grayscale information processing based on the gray value of different targets. Threshold segmentation can be divided into local threshold method and global threshold method. The global threshold method divides the image into two regions of the target and the background by a single threshold [15]. The local threshold method needs to select multiple segmentation thresholds and divide the image into multiple target regions and backgrounds by multiple thresholds.

The most commonly used threshold segmentation algorithm is the largest interclass variance method (**Otsu**) [20], which selects a globally optimal threshold by maximizing the variance between classes. In addition to this, there are entropy-based threshold segmentation method, minimum error method, co-occurrence matrix method, moment preserving method, simple statistical method, probability relaxation method, fuzzy set method, and threshold methods combined with other methods [21]. The advantage of the threshold method is that the calculation is simple and the operation speed is faster. In particular, when the target and the background have high contrast, the segmentation effect can be obtained. The disadvantage is that it is difficult to obtain accurate results for image segmentation problems where there is no significant grayscale difference or a large overlap of the grayscale values in the image [21]. Since it only takes into account the gray information of the image without considering the spatial information of the image, it is sensitive to noise and grayscale unevenness, leading it often combined with other methods [22].

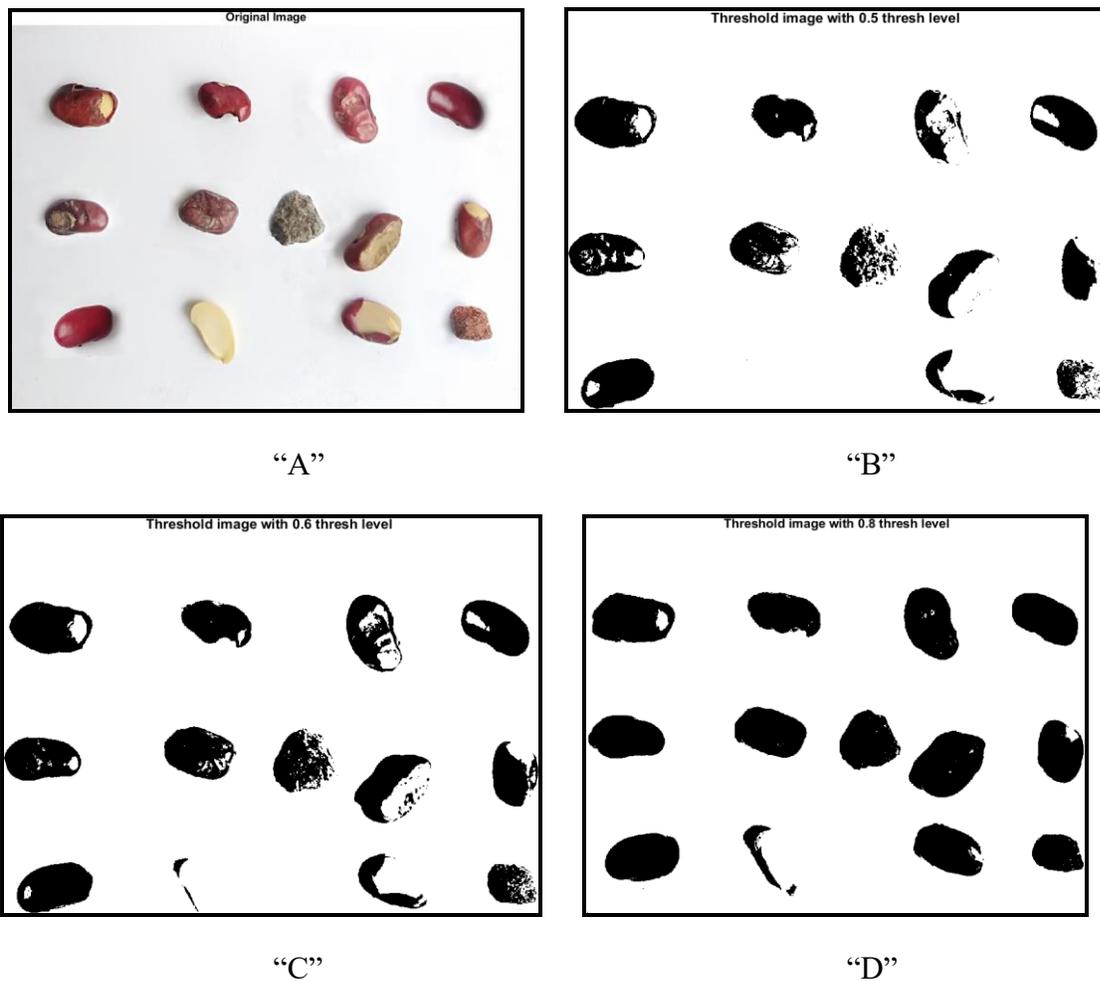


Figure 4-3-1: Result of threshold segmentation algorithm at different threshold levels; “A” Original image, “B” 0.5 Threshold level, “C” 0.6 Threshold level, “D” 0.8 Threshold level

Threshold level at 0.6 better segments image, which enable different kinds of features to be easily extracted from the RKB kernels, helps to label each kernel to their corresponding categories

Sobel Operator for edge detection

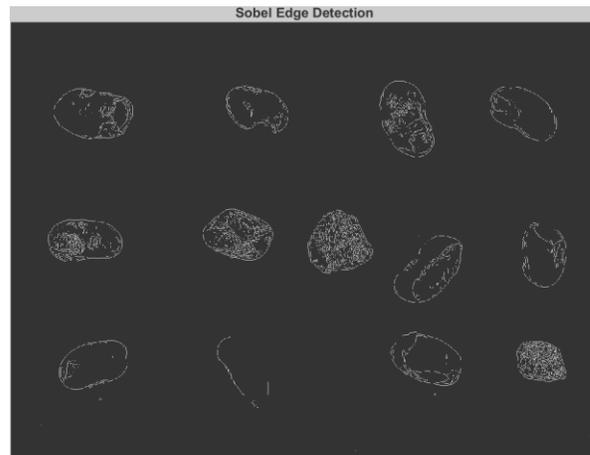
The Sobel operator is mainly used for edge detection, and it is technically a discrete differential operator used to calculate the approximation of the gradient of the image luminance function. The Sobel operator is a typical edge detection operator based on the first derivative. As a result of the operator in the introduction of a similar local average operation, the noise has a smooth effect, and can effectively eliminate the impact of noise. The influence of the Sobel operator on the position of the pixel is weighted, which is better than the Prewitt operator and the Roberts operator. The Sobel operator consists of two sets of 3x3 matrices, which are transverse and longitudinal templates, and are plotted with the image plane, respectively, to obtain the difference between the horizontal and the longitudinal difference. In actual use, the following two templates are used to detect the edges of the image.



“A”



“B”



“C”

Figure 4-3-2: Sobel edge detection algorithm; “A” Original RKB Image, “B” Gray Scale RKB image, “C” Sobel edge detected Image

As shown in Figure 4-3-2 “C” direct application of Sobel operator for edge detection contains more original image edge information but there still exists some rough edge detection, inaccurate detection due to the type of original image like the one seen in figure 4-3-2 “C” with partial visibility of edge, this image is broken RKB with the white foreground. But in general, it displayed more information of the original image; the edge is more specific and contains more information.

Selecting the best performing segmentation algorithm

Experimental results show that it is apparent to obtain better results by using global image threshold using Otsu method threshold segmentation the proposed algorithm obtains better result than sobel operator. In the proposed algorithm, it takes little time to get a precise result when small size structural operator is selected. And the continuous contour of an image is easy to obtain when the large size structural operator is selected. The proposed algorithm also has an obvious

advantage in noise restraining, which is a good edge detecting and image segmentation algorithm with wide applicability.

4.4 Test Results of Classification

Tests are conducted on naïve Bayes and ANN classifiers to determine the best performing classifier based on the criterion of classification accuracy. Hereunder experimental results are presented for the selected classification algorithms.

4.4.1 Experimental Setup

To assess the classification performance of different sets of characteristics, we created four experimental situations. That is, the categorization was tried using color, size, and form parameters, as well as a combination of them. Each of these three instances was tested using the two previously described classifiers to compare their generalization skills. They are the stages of training and testing. Data is repeatedly provided to the classifier throughout the training phase, and weights are changed to get the desired response.

During the testing step, the trained system is applied to data that it has never seen before to assess the classification's performance. As a result, we must divide the complete data set into training and testing data sets before designing the classifier. 70 percent of each region's complete data set was used to develop training, while the remaining 30 percent was used for testing data. As a result, 407 data sets were used for training and 175 for testing out of a total of 582 data sets.

For simplifying the use of the proposed model for classification and grading of Ethiopian RKB, we designed a user interface shown in figure 4-4-1. It allows preprocessing of the RKB image data and shows the class and grade of the bean for the given RKB image.

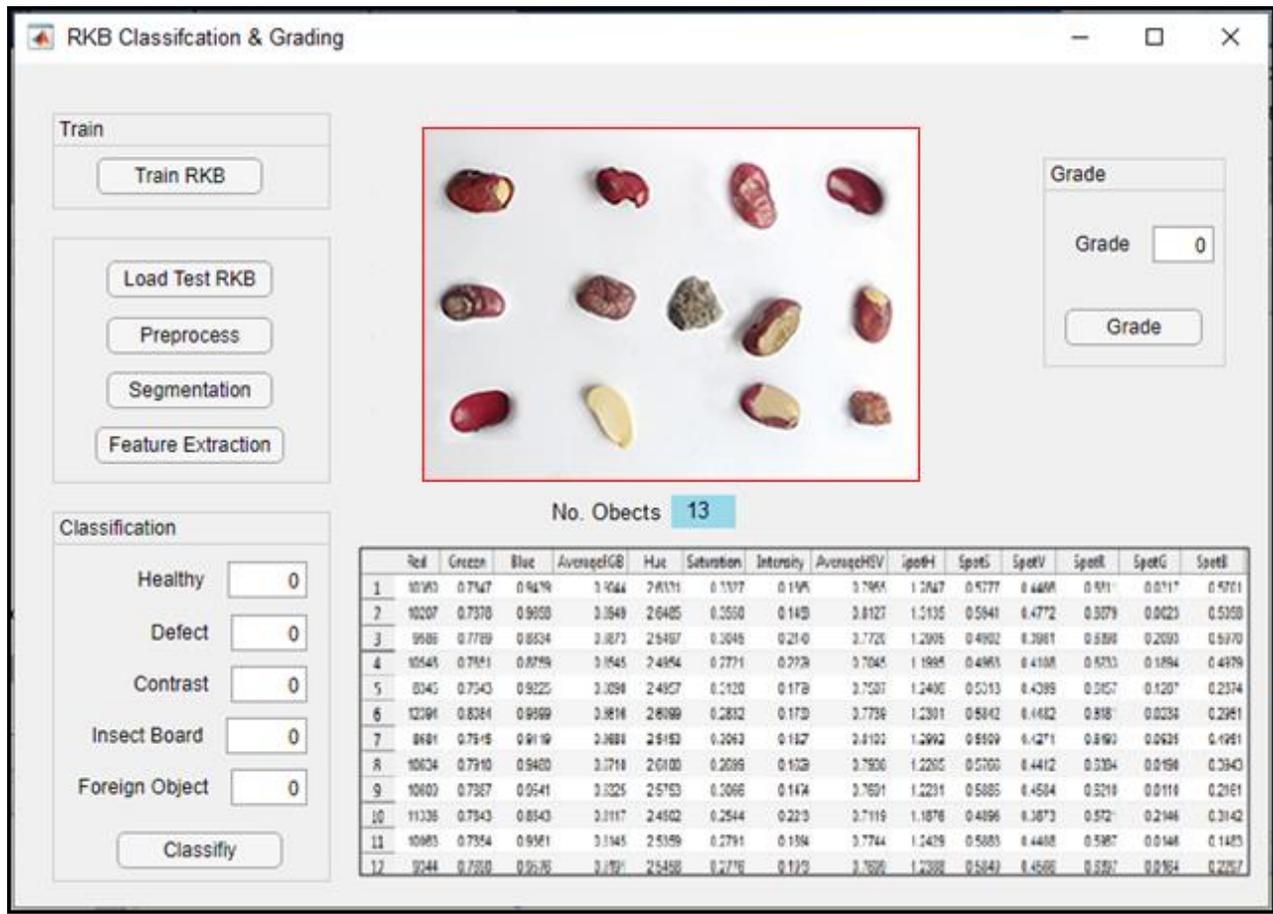


Figure 4-4-1: Graphical user interfaces

4.4.2 Naïve Bayes Classifier Test Results

The performance of the naïve Bayesian classifier was evaluated using 175 data items (30% of the data set). The result of the naïve Bayesian classifier confusion matrix is shown in Table 4.2. Instances that were accurately classified are represented by the diagonal elements. Defect, Foreign Object, Healthy, Contrast, and Insect boards have classification accuracy of 72.2 percent, 100 percent, 94.5 percent, 100 percent, and 71.4 percent, respectively, using this classifier.

Table 4-4-1: Test Confusion Matrix of Naïve Bayesian Classifier

		Target Class				
		Defect	Foreign Object	Healthy	Contrast	Insect Board
Output Class	Defect	26	0	1	0	5
	Foreign Object	0	13	0	0	0
	Healthy	3	0	69	0	1
	Contrast	2	0	0	32	0
	Insect Board	5	0	3	0	15
	Classification Accuracy	72.2%	100%	94.5%	100%	71.4%

The overall classification accuracy obtained is 89%. This is calculated by summing the number of correctly classified kernels in each class and dividing the result by the total number of test data (175). The classification accuracy for the classes Defect and Insect board is below 75%. This has affected the overall performance of the naïve Bayesian classifier to significantly underperform, compared to the neural network classifier.

4.4.3 ANN Classifier Test Results

After the data was partitioned as explained in Section 4.2, the neural network is trained. The whole process, i.e., training, validation, and testing took only 5seconds. During training, cross-entropy was used as the error function. The neural network training process is halted at the 46th iteration (epoch) at which the validation error started to rise and the training error was dropping. This training process is shown in Figure 4-4-2 below.

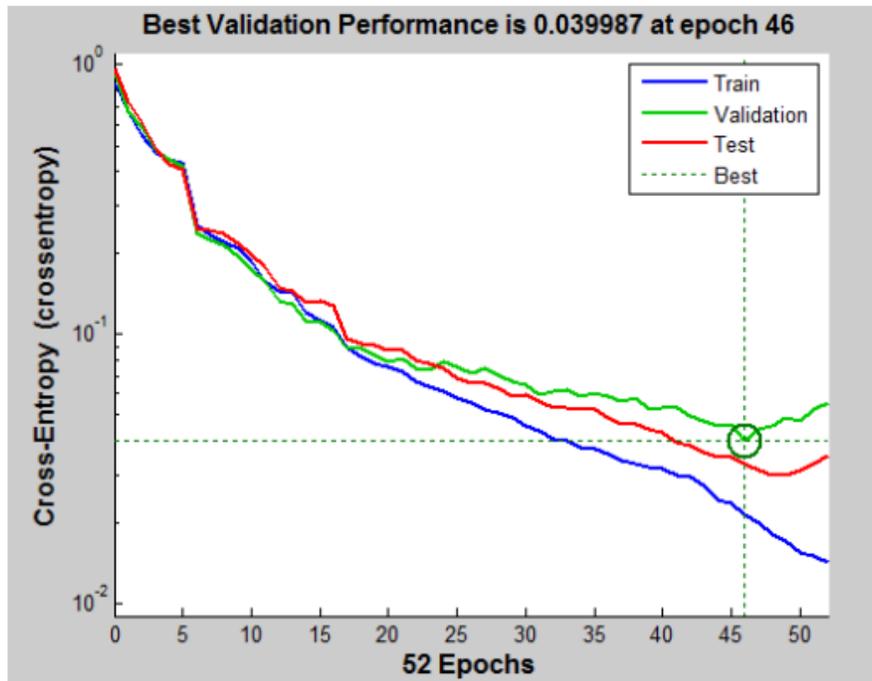


Figure 4-4-2: Cross-Entropy Error Showing the Performance of the Trained ANN

Accordingly, classification accuracies of 98.7%, 95.0%, and 96.3% have been achieved for training, validation, and testing respectively. Moreover, an overall classification accuracy of 93.8% is achieved. This accuracy is calculated by dividing the total number of correctly classified kernels by 582 (by the total number of kernels in the sample). The confusion matrix showing the overall classification results (including training, validation, and testing) is shown in Table 4-4-2.

Table 4-4-2: Confusion Matrix Showing Overall Classification Accuracy

		Target Class					
		Foreign Object	Healthy	Defect	Insect Board	Contrast	
Output Class	Foreign Object	40 6.4%	0	0	0	0	100%
	Healthy	0	200 34.3%	7	3	0	95.2%
	Defect	0	3	131 22.5%	4	5	92%
	Insect Board	0	2	5	38 6.5%	0	84.4%
	Contrast	0	5	0	0	137 23.5%	95%
		100%	95.2%	92%	84.4%	95%	93.8%

Since the naïve Bayesian classifier resulted in 89% of classification accuracy and the ANN achieved 94% for the same, we conclude that ANN outperforms the naïve Bayesian classifier. To identify which feature groups contribute the most to the classification process, we prepared four different scenarios and trained the network under each.

Table 4-4-3: Performance of model in different classifiers

Classification algorithm	Correctly classified instances	Accuracy (in %)	Running time
Naïve Bayes	516	89%	4.1
ANN	546	94%	3.8

Scenario One

In kernels, the area occupied by the damage and the associated hue, saturation, value (intensity), red, green, and blue color values are utilized to display insect board, contrast, and defect areas. In this work, the hue, saturation, value, red, green, and blue values associated with damaged areas inside a kernel are referred to as spot-hue, spot-saturation, spot-value, spot-red, spot-green, and spot-blue. In this circumstance, the usefulness of the attributes spot-hue, spot-saturation, spot-value, spot-red, spot-green, and spot-blue is studied. As a result, the classifier was retrained using training data that did not include these properties. As a result, we discovered that the discriminative strength of the spot-hue, spot-saturation, and spot-value attributes, as well as the spot-red, spot-green, and spot-blue attributes, is so strong that the ANN classifier's classification accuracy suffers significantly when they are not present. The overall classification accuracy of the ANN classifier was reduced to 93.6 percent, as shown in Table 4.3. With only 85 percent classification accuracy, the insect board class was the most impacted by the lack of these traits. To put it another way, out of a total of 45 Insect board kernels, 11 were misclassified as Insect board, 2 as healthy, 4 as a defect, 2 as broken, and 1 as discolored. Only 34 kernels were categorized accurately as Insect board. In this scenario, the second most impacted class is Healthy, which has a classification accuracy of 88%. This means that 11 healthy kernels were misclassified as a defect, 9 as insect board, and 5 as contrast, out of a total of 210 healthy kernels. Only 185 people were appropriately identified as being in good health.

Table 4-4-4: Confusion Matrix for Scenario-1

		Target Class					
Output Class		Foreign Object	Healthy	Defect	Insect Board	Contrast	
	Foreign Object	40 6.4%	0	0	0	0	100%
	Healthy	0	185 39.7%	4	2	3	95.3%
	Defect	0	11	147 25.9%	4	0	90.7%
	Insect Board	0	9	3	34 7.7%	0	75.5%
	Contrast	0	5	0	0	139 20.3%	96.5%
		100%	88%	97.4%	85%	97.8%	93.6%

Scenario Two

The discriminative strength of the size feature is assessed in this scenario. We were able to see the influence of the area by training the neural network without include the area attribute in the feature data set. To discriminate between faults and other kernels, we first used the area as a feature to describe RKB kernel size. As a result, we expected that in this situation, defect kernels would be misclassified into distinct classes. However, the classifier's accuracy declined from 93.6 % to 90.2 %.

Table 4-4-5: Confusion Matrix for Scenario-2

		Target Class					
Output Class		Foreign Object	Healthy	Defect	Insect Board	Contrast	
	Foreign Object	32	0	5	0	0	88.6%
	Healthy	0	194	4	2	0	97%
	Defect	4	8	134	5	0	90.7%
	Insect Board	0	6	8	31	2	77.4%
	Contrast	0	4	0	0	134	97.8%
		89%	94.2%	87.3%	82.2%	98.5%	90.2%

Scenario Three

In this case, we explored the discriminative efficiency of shape features. As a result, we trained an ANN without include these characteristics in the training data set. The total classification accuracy of the classifier was reduced from 93.6 % to 89.5 %.

Table 4-4-6: Confusion Matrix for Scenario-3

		Target Class					
Output Class		Foreign Object	Healthy	Defect	Insect Board	Contrast	
	Foreign Object	39	0	0	4	4	88.2%
	Healthy	0	182	8	2	5	93.3%
	Defect	0	11	128	4	0	90%
	Insect Board	0	9	3	35	2	76.2%
	Contrast	3	5	4	2	137	95.2%
		92.5%	86%	91%	75.4%	97.8%	89.5%

Scenario Four

The discriminative potential of color characteristics is investigated in this situation. This is accomplished by training the ANN without including color characteristics in the training set. As a result, we discovered that the ANN classification accuracy reduced from 93.6 % to 71.0 %.

4.5 Discussion of Result

We demonstrated that our suggested segmentation algorithm and the models we utilized to describe RKB sample attributes served their intended functions. This is demonstrated by the high degree of classification accuracy attained. Furthermore, it has been demonstrated that color features have considerably greater discriminative strength than size and shape features.

Table 4-5-1: The four scenarios discriminative efficiency

Scenarios	Correctly classified instances	Accuracy (in%)
Discriminative efficiency of Spot HSV & Spot RGB	545	93.6
Discriminative efficiency of size features	525	60.2
Discriminative efficiency of shape features	521	89.5
Discriminative efficiency of color features	412	71.0

In Scenario one, no reduction in classification accuracy is observed for the class foreign matters and the class contrast. The significant drop in classification accuracy of the classes' healthy, insect board classes is because all these kernels have spots on their surface in Figure 4-5-1 "A" and "B" respectively.



Figure 4-5-1: "A" Spots formed by healthy; "B" Spots formed by insect board

The spots on insect board kernels indicate that the insects at that location have consumed the kernels. The spots on the healthy kernels indicate that they have been subjected to insect board or defector contrast at those points. Because of their distinct color features, insect board, contrast, and defect areas inside a kernel differ from the rest of the kernel area. Insect board kernels have distinct colors at specific locations. Similarly, we've discovered that contrast and defect kernels have distinct color properties at discolouration and rottenness spots.

In scenario two, all contrast kernels (97.8%) are classified correctly without including the attribute area from the feature set is that even though it is excluded, it is still present in composite features such as average red, green, blue, hue, saturation, intensity values. This is because composite features are calculated by summing the respective color value of a kernel and dividing

it by the kernel area. Hence, the effect of the area will not be ruled out by its exclusion from the feature set. Thus, we can conclude that color features have the highest discriminative power, and the size feature has the second-highest discriminative power. However, we found out that shape features have the least discriminating power for the assessment of the RKB sample. The comparison of the discriminative power of size, color, and shape features is presented in Figure 4-5-2 as a column chart. In this chart, the discriminative power of size, color, and shape features is compared by using the observed drop in the classification accuracy during scenario 2, and scenario 3.

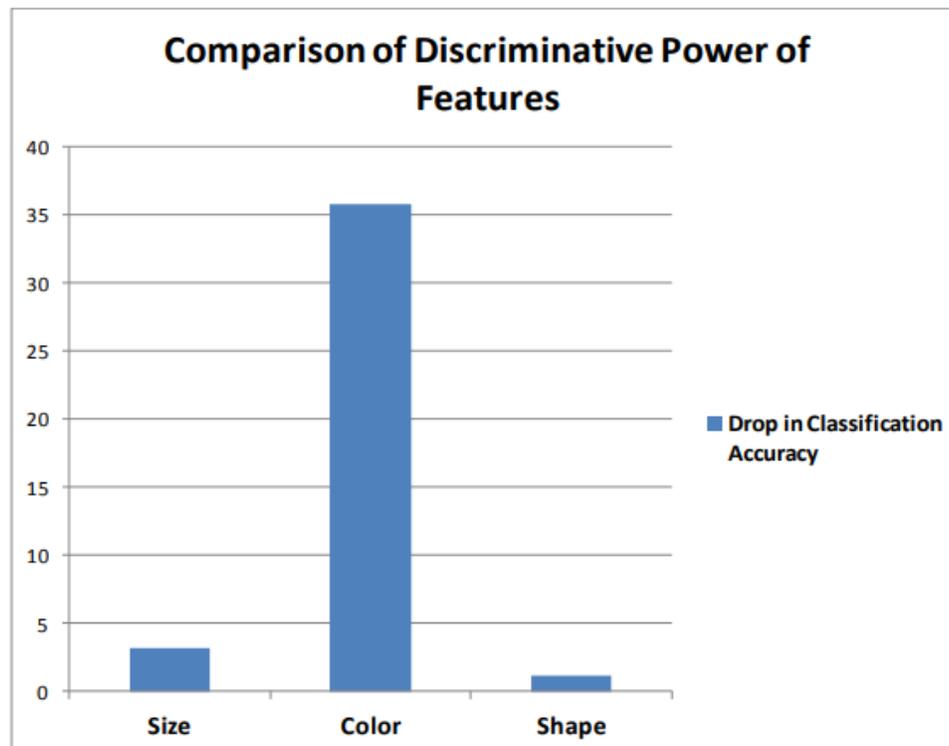


Figure 4-5-2: Comparison of Discriminative Power of Size, Shape, and Colour features

One of the challenges of this work is

- Lack of proper laboratory settings for image acquisition. In addition to this, the quality of the camera, the image acquisition environment, and other imaging factors may affect the result.
- Moreover, the number of grains in the collected RKB sample for some classes like insect board is very small. This has its effect on the achieved result.
- The other major issue is that some kernels exhibit the properties of more than one class which results in misclassification.

5.2 Contribution to Knowledge

Despite the fact that several related works on cereal grains have been carried out in the field of digital image processing, the development of classification and grading of RKB using image processing techniques for the recognition of defect, insect board, contrast, and foreign matter has not been attempted so far.

The key distinction between this work and other works is the research aims. Furthermore, the specific grain type each study is concerned with, the research area (i.e., whether it is grain variety identification or grain quality identification), and the number of 43 classes utilized in the classification process are other criteria that distinguish our work from other related works. The difference between our work and grain variety identification works is that grain variety identification deals with the identification of the different kinds(species) of a particular grain type while grain quality identification (i.e., our work) works deal with the identification of foreign, insect board, contrast, shriveled, broken and healthy constituents of RKB. However, it is important to consider the respective research objectives, the number and type of classes, and the particular grain type used in each research as criteria to distinguish our work and the other grain quality assessment works

This research work has contributed the following to the area of digital image processing in relation to automatic RKB grain quality assessment.

- We proposed a segmentation algorithm that is very well suited to segmenting RKB sample images. In the future, other grains could be added to the algorithm.
- A total of 24 features were identified that were utilized to successfully categorize objects detected in RKB sample images and assign RKBs to a grade level category.
- We develop a classifier (based on Artificial Neural Networks) that correctly classifies the objects in the RKB sample image.
- Finally, we suggested a system architecture for assessing the quality of RKB samples automatically.

CHAPTER FIVE

Conclusion and Future Work

5.1 Conclusion

RKB is one of the world's most widely grown grains. Governments and farmers invest arable land, energy, time, and money in this crop every year. Ethiopia, like many other countries, manufactures RKB for both domestic and export markets. The grain is an important food source all over the world, but especially in Sub-Saharan Africa. In developed countries, RKB is widely used as a raw material in industrial products. RKB is also used as a raw material in the processing of foods. RKB grains may be damaged during harvesting, storage, and transportation. Some types of damage simply reduce the quality of the grain, while others make it unsafe to consume. As a result, governments impose quality standards on RKB destined for either the domestic or international markets.

This standard establishes criteria for assessing RKB quality. The RKB morphological and chemical properties are used to create the standard. There is currently no automatic method for evaluating RKB quality. Rather, RKB quality is determined by hand. Manual evaluation, on the other hand, takes a long time and necessitates the use of qualified and experienced personnel. This is particularly noticeable during large-scale inspections. This manual quality evaluation approach is prone to prejudice, inconsistency, and corruption (bribery). It is critical to use an automated quality assessment system to eliminate the majority of the flaws in manual work and to lessen the associated corruption. The advantages of automated RKB quality assessment versus manual RKB quality assessment are numerous. The objective aspect of automated RKB quality assessment is its main benefit. This aids in accurately describing apparent features without prejudice or contradictions. Automated systems save time and effort when compared to manual ones. Furthermore, an automated quality assessment system eliminates the potential for corruption (bribery) that exists in a manual system and facilitates national and international trade. As a result, an automatic RKB quality evaluation system is created in this study to classify RKB samples consistently and objectively. To do so, a unique segmentation approach for identifying the damaged portions of RKB kernels is created. To simulate the elements of the RKB sample, a total of 24 characteristics were discovered. Furthermore, the proposed segmentation technique is used to construct the system architecture. A feedforward artificial neural network with 24 input nodes and 5 output nodes, as well as a backpropagation method, is developed for classification purposes, with the number of input features and output classes corresponding to the number of input features and output classes, respectively. The network's performance is compared to that of other existing classifiers, both empirically and using data from the literature as support.

The overall success percentage for the classification of the RKB sample is 93.8%, according to the results. Foreign object, flaw, healthy, contrast, and insect board kernel detection success percentages are 100%, 92 %, 95.2%, 95%, and 84.4%, respectively. Furthermore, these findings

suggest that the proposed segmentation algorithm and system architecture are effective in evaluating the quality of RKB sample constituents following the ECX standard specified for RKB samples. As a result, using digital image processing and ANN, it is possible to analyze the quality of RKB samples. As a result, the negative characteristics of manual labor can be practically eliminated. The lack of suitable laboratory circumstances for image acquisition is a limitation of this investigation. In addition, the camera's quality, the image acquisition environment, and other image quality factors must all be considered.

5.2 Future Work

Though this study has been able to assess the RKB sample quality successfully, some works require further investigation. The following are the possible future works.

- In this study, RKB sample photos are taken with a single camera. As a result, the camera only captures the RKB sample components that are facing it. For a complete automated RKB sample inspection, the system should inspect both sides of each kernel. For double-sided image collection, two cameras and clear material to lay the sample constituents could be used.
- Shadows that may have appeared during the image capture process. As a result, future research can broaden by incorporating an existing or new algorithm to remove the shadow effect.
- Include analysis of moisture content and mass determination (measuring the quantity of water in a material improves categorization accuracy). Excessive or insufficient moisture content in a substance can have a negative impact on its physical properties. Weight, thermal expansion, amalgamation, and electrical conductivity are all considerations.
- Due to a lack of sample data, the current study only looked at RKB classes. As a result, future research could include the Deep Red Kidney Bean and Brown Reddish Kidney Bean classes, which could be used to classify and grade RKB sample elements.

References

- [1] ECX. (2021). *Red kidney bean, Commodities, Ethiopian commodity exchange* [Online]. Available: [https:// www.ecx.com.et](https://www.ecx.com.et)
- [2] V.Sandeep Varma¹, K.Kanaka Durgaand Keshavulu, “Seed image analysis: its applications in seed science research”, *International Research Journal of Agricultural Sciences*, vol. 1, no--- pp. 30-36, 2013.
- [3] Raja Sekar L, Ambika N, Divya Vand Kowsalya T, Assistant Professor, UG Scholar, “Fruit Classification System Using Computer Vision: A Review”, *International Journal of Trend in Research and Development*, Volume 5(1), ISSN: 2394-9333.
- [4] Teklehaymanot Girma, Sudhir Kumar Mohapatra, Befkadu Belete “Feature Extraction and Classification of Green Mung Bean Using Machine Learning”, MSc Thesis, Addis Ababa Science and Technology University, Addis Ababa University, 2019
- [5] Raji A. O, A. O. Alamutu, "Prospects of Computer Vision Automated Sorting Systems in Agricultural Process Operations in Nigeria," *The CIGR Journal of Scientific Research and Development*, vol. Vol. VII, February 2005.
- [6] Xu Liming, Zhao Yanchao, "Automated strawberry grading system based on image processing," September 2009.
- [7] E. Asfaw, "Physical Quality and Grading Systems of Ethiopian Coffee in Demand-Supply Chain", *Four Decades of Coffee Research and Development in Ethiopia*, 2007.
- [8] D.-W. S. TadhgBrosnan, "Improving quality inspection of food products by computer vision - a review," *Journal of Food Engineering*, 6 May 2003.
- [9] B. Tudu, "An Automated Machine Vision Based System for Fruit Sorting and Grading," 2012.
- [10] Mesfin Fekadu Abeza, “Development of an automated grading system of white pea beans using image processing techniques convergence with ANN”, Master’s Thesis, Addis Ababa Science and Technology University, Unpublished Master’s Thesis, 2019.
- [11] Hiwot Desta Alemayehu “Development of Automatic Sesame Grain Classification and Grading System Using Image Processing Techniques”, MSc Thesis, Addis Ababa university college of natural sciences, Addis Ababa, Ethiopia, 2017.
- [12] Daniel Hailemariam Lemessa, “Development of Automatic Maize Quality Assessment System Using Image Processing Techniques”, MSc Thesis, Department of computer science, Addis Ababa university, Unpublished Master’s Thesis, October 2015.

- [13] Asma Redi Baleker, “Raw Quality Value Classification of Ethiopian Coffee biased on Ethiopian Commodity Exchange (ECX)”, MSc Thesis, Department of computer science, Addis Ababa University, Unpublished Master’s Thesis, November 2011.
- [14] Darci J. Harland. (2021). *An Introduction to Experimental Research* [Online]. Available: <https://cemast.illinoisstate.edu>
- [15] S. Saini and K. Arora, “A Study Analysis on the Different Image Segmentation Techniques”, *International Journal of Information and Computation Technology*, Vol. 4, No. 14, pp. 1445-1452, 2014.
- [16] R. Gonzalez and R. Woods, “Digital Image Processing”, Third Edition, Addison Wesley, 2008.
- [17] World Food program, “Technical Specifications for ETHIOPIA HARICOT BEANS”, V1.0 30, May 2011.
- [18] T. Acharya and A. Ray, *Image Processing Principles and Applications*, John Wiley, 2005.
- [19] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing*, First Edition, Wiley-Blackwell, 2011.
- [20] R. Dass, Priyanka, and S. Devi, “Image Segmentation Technique”, *International Journal of Electronics and Communication Technology*, Vol. 3, Issue 1, pp. 6670, January-March 2012.
- [21] A.Ashraf, B.Safaai and Z.Nazar, “Research review for digital segmentation techniques”, *International Journal of Computer Science & Information Technology (IJCSIT)*, vol.3, October 2011.
- [22] R.Yogamangalam and B.Karthikeyan, “Segmentation Techniques Comparison in Image Processing”, *International Journal of Engineering and Technology (IJET)*, School of Computing, University of Thanjavur, India, 2016.
- [23] P.Sharma, S.Gurpreet and K.Amandeep, “Different Techniques Of Edge Detection In Digital Image Processing”, *International Journal of Engineering Research and Applications (IJERA)*, vol. 3, pp. 458-461, May-June 2013.
- [24] Coolen, A. C. C., R. Kühn and P. Sollich: *Theory of Neural Information Processing Systems*, Oxford University Press, 2005.
- [25] William K. Pratt, “Digital image processing”, PIKS Scientific inside, John Wiley, 4th Edition, 2007.
- [26] John C. Russ, *The Image Processing Handbook*, 3rd Edition, CRC Press, 1998.

- [27] Majumdar, S. and D.S. Jayas, “Classification of cereal grains using machine vision. IV. Combined morphology, color and texture models”, *International Journal Trans. ASAE*, 43(6):1689-1694, 2000.
- [28] Granitto, P. M., H. D. Navone, P. F. Verdes and H. A. Ceccatto, “Automatic Identification of Weed Seeds by Color Image Processing”, *Instituto de Fisica Rosario (CONICET-Universidad Nacional de Rosario)*, Argentina, 2000.
- [29] Zayas, I., Y. Pomeranz and F.S. Lai, “Discrimination of Wheat and Non-wheat Components in Grain Samples by Image Analysis”, *Journal of Cereal Chemistry*, 6(3):233-237, 1989
- [30] Liu, Zhao-yan, Cheng Fang, Ying Yi-bin and Rao Xiu-qin, “Identification of Rice Varieties Using Neural Networks”, *Journal of Zhejiang University Science*, 6B (11):1095-1100, 2005.
- [31] Visen, N.S., J. Paliwal, D.S. Jayas, and N.D.G. White, “Specialist Neural Networks for Cereal Grain Classification”, *Bio systems engineering*, 82 (2): 151–159, 2002.
- [32] Jayas, D. S., J. Paliwal and N. S. Visen, “Multi-layer Neural Networks for Image Analysis of Agricultural Products”, *Journal of Agricultural Engineering Research*, 77(2):119-128, 2000.
- [33] Ding, K. and S. Gunasekaran, “Shape feature extraction and Classification of food Material Using Computer Vision”, *Journal of Food and Process Engineering Institute of ASAE*, 1994.
- [34] S. Arun Balaj and K. Baskaran, “Design and Development of Artificial Neural Networks (ANN) to Predict Annual Rice Production”, *International Journal of Computer Science, Engineering and Information Technology (IJCSSEIT)*, Vol. 3, No.1, pp. 13-31, 2013.
- [35] B. Raba, K. Nowakowski, and P. Boniecki, “Visual Quality Evaluation of Malting Barley with use of Neural Image Analysis”, *Journal of Research and Applications in Agricultural Engineering*, Vol. 60, No. 1, pp. 80-83, 2015.
- [36] A. Pazoki and Z. Pazoki, “Classification System for Rain Fed Wheat Grain Cultivars Using Artificial Neural Network”, *African Journal of Biotechnology*, Vol. 10, No. 41, pp. 8031-8038, 2011.
- [37] C. Lira and P. Pina, “Automated Grain Shape Measurement Applied to Beach Sands”, *Journal of Coastal Research SI*, Vol. 56, pp. 1527-1531, 2009.
- [38] B. Armstrong, “Using Digital Image Analysis for Assessing the Quality of Wheat and Barley”, Master’s Thesis, School of Science and Engineering, University of Ballarat, December 2004.

- [39] J. Han, M. Kamber, and J. Pei, "Data Mining Concepts and Techniques", Third Edition, *Morgan Kaufmann Publishers*, 2012
- [40] A.Vibhute and S.Bodhe, "Applications of Image Processing in Agricultural", *International Journal of Computer Applications*, vol. 52, August 2012.
- [41] Saravanan K and S. Sasithra, "Review on Classification based on Artificial Neural Networks", *International Journal of Ambient Systems and Applications (IJASA)*, Vol.2, December 2014.
- [42] Amit Ganatra, Y P Kosta, Gaurang Panchal and Chintan Gajjar, "Initial Classification Through Back Propagation In a Neural Network Following Optimization Through Genetic Algorithm", *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol 3, February 2011.
- [43] Zayas I., Pomeranz Y., and F. S. Lai, "Discriminate between wheat and non-wheat components in a grain sample", *Cereal Chemistry*, vol. 66, no.3, 1989.
- [44] Lai FS, Zayas I, Pomeranz Y., "Application of pattern recognition techniques in the analysis of cereal grains", *Cereal Chemistry*, Vol. 63, No. 2, pp. 168–174, 1986.
- [45] N. S. Visenl, J. Paliwall, D. S. Jayas¹, and N. D. G.White, "Image analysis of bulk grain samples using neural networks", *Can. BioSyst.Eng*, vol. 46, 2004.
- [46] B. S. Anami, D. G. Savakar, Aziz Makandar, and P.H. Unki, "A neural network model for classification of bulk grain samples based on color and texture", *Proceedings of International Conference on Cognition and Recognition*, 2005.
- [47] H.K. Mebatsion, J. Paliwal, D.S. Jayas, "Automatic classification of non-touching cereal grains in digital images using limited morphological and color features", *Computers and Electronics in Agriculture*, 2013.
- [48] D.Kruzlicova, J.Mocak, E.Katsoyannos and E.Lankmayr, "Classification and characterization of Olive Oils Using Artificial Neural Network", *International Journal of Food and Nutrition Research*, Vol. 47, pp. 181–188, 2008.
- [49] A. Pazoki and Z. Pazoki, "Classification System for Rain Fed Wheat Grain Cultivars Using Artificial Neural Network", *African Journal of Biotechnology*, Vol. 10, No. 41, pp. 8031-8038, 2011.
- [50] L.W.Steenhoek, M.K.Misra, C.R.Hurburgh.Jr and C.J.Bern,"Implementinga computer vision system for corn kernel damage evaluation," *Applied Engineering in Agriculture*, vol. 17, no. 2, p. 235, 2001.

- [51] Habtamu Minassie Aycheh, “Image Analysis for Ethiopian Coffee Classification”, unpublished Master’s Thesis, School of Graduate Studies, Addis Ababa University, January 2008.
- [52] Tom Patten, Wenjing Li, George Bebis and Muhammad Hussain, “assessing insect growth using image analysis”, *International Journal on Artificial Intelligence Tools*, Vol. 20, No. 03, pp. 511-530, 2011.
- [53] A. Soleimanipour, G. R. Chegini, J. Massah, and P. Zarafshan, “A novel image processing framework to detect geometrical features of horticultural crops: case study of anthurium flowers,” *Scientia Horticulturae*, vol. 243, pp. 414–420, 2019.
- [54] Salem, S.A., “Image Segmentation by Using Edge Detection”, *International Journal on Computer Science and Engineering*, Vol. 02, No. 03, 804-807, 2010.
- [55] Pavlidis, T, “Algorithms for Graphics and Image Processing”, Computer Science Press, Rockville, MD, 1982.
- [56] Rocky Sir. (2021). *Suven Consultants & technology Pvt. Ltd* [Online]. Available: <https://datascience.suvenconsultants.net>
- [57] C. Oprea, “Performance evaluation of the data mining classification methods”, *Information society and sustainable development*, University of Târgu Jiu, Vol 2, April 2014.
- [58] Afshin Gholamy, Vladik Kreinovich, Olga Kosheleva, “Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation”, Department of Computer Science, The University of Texas at El Paso, 2018.
- [59] Sheetal Sharama. (Aug 8, 2017). *Artificial Neural Network (ANN) using Machine learning*, [Online]. Available: <https://www.datasciencecentral.com/artificial-neural-network-ann-in-machine-learning/>
- [60] Matworks. (2021). *Machine learning in MATLAB*. [Online]. Available: <https://www.mathworks.com/discovery/machine-learning.html>
- [61] Goholamreza Anbarjafari. (2021). *Introduction to image processing* [Online]. Available: <https://sisu.ut.ee/imageprocessing/book/1>

Appendices A - System Code

```
function varargout=RKBQuality(varargin)
% RKBQUALITYMATLABcodeforRKBQuality.fig
% RKBQUALITY,byitself,createsanewRKBQUALITYorraisesthe existing
% singleton*.

%H=RKBQUALITYreturnsthehandletoanewRKBQUALITYorthe handle to
%theexistingsingleton*.
% RKBQUALITY('CALLBACK',hObject,eventData,handles,...)callsthe local
% functionnamedCALLBACKinRKBQUALITY.Mwiththegiveninputarguments.
%
% RKBQUALITY('Property','Value',...)createsanewRKBQUALITYor
%raisesthe
%existingsingleton*.Startingfromtheleft,propertyvaluepairsare
%appliedtotheGUIbeforeRKBQUALITY_OpeningFcngets called.An
%unrecognizedpropertynameorinvalidvaluemakespropertyapplication
%stop.AllinputsarepassedtoRKBQuality_OpeningFcnvia varargin.

%*SeeGUIOptionsonGUIDE'sToolsmenu.Choose"GUIallowsonlyone
%instancetorun(singleton)".
%
%Seealso:GUIDE,GUIDATA,GUIHANDLES
%EdittheabovetexttomodifytheresponsetohelpRKBQuality
%LastModifiedbyGUIDEv2.529-Oct-201707:21:30
% Begin initialization code - DO NOT
EDITgui_Singleton= 1;
gui_State=struct('gui_Name',mfilename,...
                'gui_Singleton',gui_Singleton,
                ...'gui_OpeningFcn',@RKBQuality_OpeningFcn,...'gui_
                i_OutputFcn',@RKBQuality_OutputFcn,
                ...'gui_LayoutFcn',[], ...
                'gui_Callback',
                []);

if nargin&&ischar(varargin{1})
    gui_State.gui_Callback=str2func(varargin{1});
end

if nargin
[varargout{1:nargout}]=gui_mainfcn(gui_State,varargin{:});else
    gui_mainfcn(gui_State,varargin{:});
end
%Endinitializationcode-DONOTEDIT
% --- Executes just before RKBQuality is made visible.
FunctionRKBQuality_OpeningFcn(hObject,eventdata,handles,varargin)
%Thisfunctionhasnooutputargs,seeOutputFcn.
%hObject handletofigure
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles structurewithhandlesanduserdata(seeGUIDATA)
%varargin commandlineargumentstoRKBQuality(seeVARARGIN)
%ChosedefaultcommandlineoutputforRKBQualityhandles.output=
hObject;
```

```

% Update handles
structureguidata(hObject,handles);
%UIWAITmakesRKBQualitywaitforuserresponse(seeUIRESUME)
%uiwait(handles.figure1);
% --- Outputs from this function are returned to the command
line.functionvarargout = RKBQuality_OutputFcn(hObject,
eventdata,handles)
%varargoutcellarrayforreturningoutputargs(seeVARARGOUT);
%hObject handletofigure
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles structurewithhandlesanduserdata(seeGUIDATA)

%Getdefaultcommandlineoutputfromhandlesstructurevarargout{1}
= handles.output;

%---ExecutesonbuttonpressinLoadimage.
functionLoadimage_Callback(hObject,eventdata,handles)
%hObject handletoLoadimage(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles
structurewithhandlesanduserdata(seeGUIDATA)handles.output=hO
bject;
[a
b]=uigetfile({'*.*'});img=imread([b
a]);handles.rgbimg=img;imshow(i
mg,'Parent',handles.axes1);guida
ta(hObject,handles);
%---ExecutesonbuttonpressinPreprocessimage.
functionPreprocessimage_Callback(hObject,eventdata,handles)
%hObject handletoPreprocessimage(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles structure with handles and user data (see
GUIDATA)handles.output=hObject;
tic
I =
handles.rgbimg;imgGray=rgb
2gray(I);
%imgGray=imclearborder(imgGr)
;gra=medfilt2(imgGray,[1111])
;
L=graythresh(gra);%find appropriate gray thresh
valueBW1=im2bw(gra,L);%convert to
binarybw2=imfill(BW1,'holes');filterim=bwareaopen(bw2,35
0);%opensareagreaterthan350[~,thresh]=edge(filterim,'sob
el');b=edge(filterim,'canny',thresh);
bi=im2double(b);
bwf=edge(bi,'roberts');
%figure,imshow(bw),title('scr');
bwcon=bwconncomp(filterim,8);%Findconnectedcomponentsinbinaryimagenum=bwcon.
NumObjects;
set(handles.text2,'string',num
);s=regionprops(bwf,{...
'Centroid',...

```

```

        'Area',... 'MajorAxisLength',... 'MinorAxisLength',... 'Orientation'});
%figure;
imshow(bwf); imshow(bwf, 'Parent', handles.axes2); t=linspace(0, 2*pi, 70);
hold on
for k=1:length(s)
    a =
        s(k).MajorAxisLength/2; b =
        s(k).MinorAxisLength/2; Xc=s(k).Centroid(1);
        Yc=s(k).Centroid(2); phi=deg2rad(-s(k).Orientation);
        x = Xc + a*cos(t)*cos(phi) - b*sin(t)*sin(phi); y = Yc + a*cos(t)*sin(phi)
        + b*sin(t)*cos(phi); plot(x, y, 'r', 'Linewidth', 1)
end
f=getframe(handles.axes2); %# Capture the current
windowimwrite(f.cdata, 'demo.png');

hold off
% -----
-
segmentationP
im=imread('demo.png');
Im =imresize(im, [470 680]); cform=makecform('srgb21ab'); J=applycform(Im, cform);
%figure, imshow(J), title('Colorform'); K=J(:,:,1);
%figure, imshow(K), title('equalisebrightness'); gra=medfilt2(J(:,:,1), [33]);
L=graythresh(gra(:,:,1)); %find appropriate graythresh value im=im2bw(J(:,:,1), L);
%convert to binary
%figure, imshow(im), title('Black and White'); bw2=imfill(im, 'holes');
%figure, imshow(bw2), title('hole filled');
imggry=~im2bw(bw2);
% figure,
imshow(imggry), title('change'); param=
readparam();
stats=mia_particles_segmentation(imggry, param);
%
MinorAxis=
zeros(num, 1); MajorAxis=
zeros(num, 1); Area=
zeros(num, 1); SolidityB=
zeros(num, 1); EccentricityB=
zeros(num, 1); Perimeter=
zeros(num, 1); Elongation=
zeros(num, 1); AspectRatio=
zeros(num, 1); Roundness=
zeros(num, 1); Compactness=
zeros(n

```

```

um,1);Red=zeros(num,1);Green=zeros(num,1);Blue=zeros
(num,1);
RGBimg=
im2double(I);Hue=
zeros(num,1);Satur=
zeros(num,1);Inten=ze
ros(num,1);
Homogeneity=zeros(num,1);C
orrelation=zeros(num,1);Co
ntrast=zeros(num,1);
%waitBar
%hWaitBar=waitbar(0,'Preprocessing');M
RKB=regionprops(bwcon,'all');
fori=1:num
Area(i)=MRKB(i).Area;Perimeter(i)=MRKB(i).Perimete
r;MinorAxis(i)=MRKB(i).MinorAxisLength;MajorAxis(i
)=MRKB(i).MajorAxisLength;Elongation(i)=(MinorAxis
(i)/MajorAxis(i));AspectRatio(i)=(MajorAxis(i)/Min
orAxis(i));Roundness(i)=4*pi*Area(i)/((Perimeter(i))
^2);Compactness(i)=(Perimeter(i))^2/Area(i);Solidi
tyB(i)=MRKB(i).Solidity;EccentricityB(i)=MRKB(i).E
ccentricity;SRKB=
false(size(RGBimg));SRKB(bwcon.PixelIdxList{i})=tr
ue;
R=SRKB(:,:,1);
G=SRKB(:,:,2);
B=SRKB(:,:,3);
%conversion
num=0.5*((R-G)+(R-B));
den=sqrt((R-G).^2+(R-B).*(G-B));
theta=acos(num./(den+eps));ifB<=G
H=theta;else
H=360-theta;
end
num = min(min(R, G),
B);den=R+G+B;den(den==0)=ep
s;

end

S = 1 - 3.*
num./den;H(S== 0)= 0;

I=(R+G+B)/3;

Red(i)=
mean2(R);Green(i) =
mean2(G);Blue(i) =
mean2(B);Hue(i) =
mean2(H);Satur(i) =
mean2(S);Inten(i)=mean2(
I);

SMRKBgrain=false(size(imgGray));SMRKBgrain(bwcon
.PixelIdxList{i}) =
true;glcm=graycomatrix(SMRKBgrain);

```

```

MBTexture=graycoprops (glcm, {'Homogeneity', 'Correlation', 'Contrast'}); Homogeneity(i)=MBTexture.Homogeneity; Correlation(i)=MBTexture.Correlation;
Contrast(i)=MBTexture.Contrast;
%waitbar(i/num); Featuredata_texture=[RedGreenBlueHueSaturIntenHomogeneityCorrelationContrast];
texturee=mapstd(Featuredata_texture); %processes matrices
bytransformingthemeanandstandarddeviationofeachrowtoymeanandystd
%display(texturee);
Extracted_Features=[MinorAxisMajorAxisPerimeterArea
AspectRatioElangationCompactnessRoundnessSolidityBEccentricityBtexturee];
%savetempmorpholoy.matExtracted_Features
xlswrite('RKBfeatures.xls',Extracted_Features,1);
diintable=handles.uitable1;set(diintable,'Data',Extracted_Features);
%msgbox('PreprocessingCompleted','Preprocessing');
%handles.Ex_Features=Extracted_Features;
%delete(hWaitBar);set(handles.pushbutton
3,'Enable','on');toc

guidata(hObject,handles);

functionGrade1_Callback(hObject,eventdata,handles)
%hObjecthandletoGrade1(seeGCBO)
%eventdatareserved-tobefinedinafutureversionofMATLAB
%handlesstructurewithhandlesanduserdata(seeGUIDATA)
%Hints:get(hObject,'String')returnscontentsofGrade1astext
%str2double(get(hObject,'String'))returnscontentsofGrade1asadouble
% --- Executes during object creation, after setting all
properties.functionGrade1_CreateFcn(hObject,eventdata,handles)
%hObject handletoGrade1(seeGCBO)
%eventdatareserved-tobefinedinafutureversionofMATLAB
%handles empty-handlesnotcreateduntilafterallCreateFcnscaled

%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
% SeeISPCandCOMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
functionGrade2_Callback(hObject,eventdata,handles)
%hObject handletoGrade2(seeGCBO)
%eventdatareserved-tobefinedinafutureversionofMATLAB
%handles structurewithhandlesanduserdata(seeGUIDATA)

%Hints:get(hObject,'String')returnscontentsofGrade2astext
%str2double(get(hObject,'String'))returnscontentsofGrade2asadouble
%---Executesduringobjectcreation,aftersettingallproperties.
functionGrade2_CreateFcn(hObject,eventdata,handles)
%hObject handletoGrade2(seeGCBO)
%eventdatareserved-tobefinedinafutureversionofMATLAB
%handles empty-handlesnotcreateduntilafterallCreateFcnscaled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
% SeeISPCandCOMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

function Grade3_Callback(hObject,eventdata,handles)
%hObject      handletoGrade3 (seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata (seeGUIDATA)
%Hints:get (hObject,'String') returnscontentsofGrade3astext
%str2double (get (hObject,'String')) returnscontentsofGrade3asadouble
%---
Executesduringobjectcreation,aftersettingallproperties.functionGrade3_C
reateFcn(hObject,eventdata,handles)
%hObject      handletoGrade3 (seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscalled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
%
%      SeeISPCandCOMPUTER.
if ispc && isequal (get (hObject,'BackgroundColor'),get
(0,'defaultUiControlBackgroundColor'))
set (hObject,'BackgroundColor','white');
end
function Undergraded_Callback(hObject,eventdata,handles)
%hObject      handletoUndergraded (seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata (seeGUIDATA)

%Hints:get (hObject,'String') returnscontentsofUndergradedastext
% str2double (get (hObject,'String')) returnscontentsofUndergradedasa double
% --- Executes during object creation, after setting all
properties.functionUndergraded_CreateFcn(hObject,eventdata,handles)
%hObject      handletoUndergraded (seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscalled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
%
%      SeeISPCandCOMPUTER.
if ispc && isequal (get (hObject,'BackgroundColor'),get
(0,'defaultUiControlBackgroundColor'))
set (hObject,'BackgroundColor','white');
end
function pure_Callback(hObject,eventdata,handles)
%hObject      handletopure (seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata (seeGUIDATA)
%Hints:get (hObject,'String') returnscontentsofpureastext
% str2double (get (hObject,'String')) returnscontentsofpureasadouble
%---
Executesduringobjectcreation,aftersettingallproperties.functionpure_Create
Fcn(hObject,eventdata,handles)
%hObject      handletopure (seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscalled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
%
%      SeeISPCandCOMPUTER.
if ispc && isequal (get (hObject,'BackgroundColor'),get
(0,'defaultUiControlBackgroundColor'))

```

```

set(hObject,'BackgroundColor','white');end
functionFood_Callback(hObject,eventdata,handles)
%hObject      handletoFood(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata(seeGUIDATA)
%Hints:get(hObject,'String')returnscontentsofFoodastext
% str2double(get(hObject,'String'))returnscontentsofFoodasadouble
%---
Executesduringobjectcreation,aftersettingallproperties.functionFood_Cre
ateFcn(hObject,eventdata,handles)
%hObject      handletoFood(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscalled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
% SeeISPCandCOMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
functionForeign_Callback(hObject,eventdata,handles)
%hObject      handletoForeign(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata(seeGUIDATA)
%Hints:get(hObject,'String')returnscontentsofForeignastext
% str2double(get(hObject,'String'))returnscontentsofForeignasadouble
% --- Executes during object creation, after setting all
properties.functionForeign_CreateFcn(hObject,eventdata,handles)
%hObject      handletoForeign(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscalled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
% SeeISPCandCOMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

functionDiscolored_Callback(hObject,eventdata,handles)
%hObject      handletoDiscolored(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata(seeGUIDATA)
%Hints:get(hObject,'String')returnscontentsofDiscoloredastext
% str2double(get(hObject,'String'))returnscontentsofDiscoloredasa double
%---
Executesduringobjectcreation,aftersettingallproperties.functionDiscolored_
CreateFcn(hObject,eventdata,handles)
%hObject      handletoDiscolored(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscalled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
% SeeISPCandCOMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),get
(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```

```

function Broken_Callback(hObject,eventdata,handles)
%hObject      handletoBroken(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata(seeGUIDATA)
%Hints:get(hObject,'String')returnscontentsofBrokenastext
%str2double(get(hObject,'String'))returnscontentsofBrokenasadouble
Executesduringobjectcreation,aftersettingallproperties.functionBroken_Crea
teFcn(hObject,eventdata,handles)
%hObject      handletoBroken(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscaled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
% SeeISPCandCOMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),get
(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
function Wrinkled_Callback(hObject,eventdata,handles)
%hObject      handletoWrinkled(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata(seeGUIDATA)
%Hints:get(hObject,'String')returnscontentsofWrinkledastext
%str2double(get(hObject,'String'))returnscontentsofWrinkledasadouble
%---
Executesduringobjectcreation,aftersettingallproperties.functionWrinkled_Cr
eateFcn(hObject,eventdata,handles)
%hObject      handletoWrinkled(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      empty-handlesnotcreateduntilafterallCreateFcnscaled
%Hint:editcontrolsusuallyhaveawhitebackgroundonWindows.
% SeeISPCandCOMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),get
(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

%---Executesonbuttonpressinpushbutton6.
function pushbutton6_Callback(hObject,eventdata,handles)
%hObject      handletopushbutton6(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handles      structurewithhandlesanduserdata(seeGUIDATA) handles
.output=hObject;
feat = handles.G_Features;
[Gradeex]=size(feat);
%disp(features);Grade1=0;
Grade2=0;
Grade3=0;
Undergrade =
0;for i=1:Gradeex
    if(feat(i,1)==1)
        Grade1 = Grade1 +
        1;elseif(feat(i,1)==2
    )

```

```

        Grade2 =Grade2 +
        1;elseif(feat(i,1)==3)
        Grade3 = Grade3 +
        1;elseif(feat(i,1)==4)
            Undergrade=Undergrade+1;
        end
    end
end
set(handles.Gradel1, 'String',
Gradel1);set(handles.Grade2, 'String',
Grade2);set(handles.Grade3, 'String',
Grade3);set(handles.Undergraded, 'String',Undergrade);
guidata(hObject,handles);
%---Executesonbuttonpressinpushbutton5.
functionpushbutton5_Callback(hObject,eventdata,handles)
%hObjecthandletopushbutton5(seeGCBO)
%eventdatareservedtobedefinedinafutureversionofMATLAB%handles
structurewithhandlesanduserdata(seeGUIDATA)handles.output=hObje
ct;
Input = xlsread('Book1.csv', 1,
'A1:A160');Target = xlsread('Book1.csv',
1, 'C1:F160');input=Input';
target=Target';x=input;t=target;
%setdemorandstream(491218382)net =
patternnet(25);view(net)
net.divideParam.trainRatio=70/100;
net.divideParam.valRatio =
15/100;net.divideParam.testRatio=15/100;
%TrainANN
[net,tr] =
train(net,x,t);nntraint
ool
% Test the
Networkoutputs=net
(input);
errors =
gsubtract(target,outputs);performance=
perform(net,target,outputs);disp(outputs);
%plotperform(tr);fig
ure,plotperform(tr)
figure,plottrainstate(tr)
figure,
plotconfusion(target,outputs)figure,
ploterrhist(errors)save('Grade_netwo
rk.mat','net')
inputs = xlsread('Book1.csv', 1,
'A1:A160');targets = xlsread('Book1.csv',
1,
'A1:A160');groupG=xlsread('Book1.csv',1,'B
1:B160');
%function
Grade=knnclassify(inputs,targets,groupG;
%disp('Result:');

```

```

%disp(Grade);handles.G_F
eatures=Grade;
set(handles.pushbutton6, 'Enable', 'on');guidata(hObject,handles);
%---Executesonbuttonpressinpushbutton3.
functionpushbutton3_Callback(hObject,eventdata,handles)
%hObjecthandletopushbutton3(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handlesstructurewithhandlesanduserdata(seeGUIDATA)handles.o
utput=hObject;
Input = xlsread('allfeatures.csv', 1,
'A1:Q117');Target = xlsread('allfeatures.csv',
1, 'S1:X117');input=Input';
target=Target'
;x=input;t=ta
rget;
%setdemorandstream(4912183
82)net=patternnet(12);
net.divideParam.trainRatio=70/100;
net.divideParam.valRatio =
15/100;net.divideParam.testRatio=1
5/100;view(net)
%trainANN
[net,tr] =
train(net,x,t);nntraint
ool
% Test the
Networkoutputs=net
(input);
errors =
gsubtract(target,outputs);performance=
perform(net,target,outputs);disp(outpu
ts);
%plotperform(tr);fig
ure,plotperform(tr)
figure,plottrainstate(tr)
figure,
plotconfusion(target,outputs)figur
e,
ploterrhist(errors)save('Class_net
work.mat','net')
inputs = xlsread('allfeatures.csv', 1,
'A1:Q117');targets = xlsread('allfeatures.csv',
1,
'A1:Q117');groupC=xlsread('allfeatures.csv',1,'R
1:R117');
%
functionclass=knnclassify(inputs,targets
,groupC);
%disp('Result:');

```

```

%disp(groupC);handles.Ex_Features=cla
ss;set(handles.pushbutton4,'Enable','on
');guidata(hObject,handles);
%---Executesonbuttonpressinpushbutton4.
functionpushbutton4_Callback(hObject,eventdata,handles)
%hObjecthandletopushbutton4(seeGCBO)
%eventdatareserved-tobedefinedinafutureversionofMATLAB
%handlesstructurewithhandlesanduserdata(seeGUIDATA)handles.o
utput=hObject;
features=handles.Ex_Features
;[Classy]=size(features);
%disp(features)
;pureRKB=0;food
B=0;
Foreign=0;
Discolored=0;
Broken=0;
wrinkled
=0;fori=1:
Class
    if(features(i,1)==1)
    pureRKB=pureRKB+1;elseif(fe
atures(i,1)==2)
    foodB =foodB+
    1;elseif(features(i,1)==3)
    Foreign = Foreign +
    1;elseif(features(i,1)=
    =4)
    Discolored=Discolored+1;elsei
f(features(i,1)==5)
Broken=Broken+1;

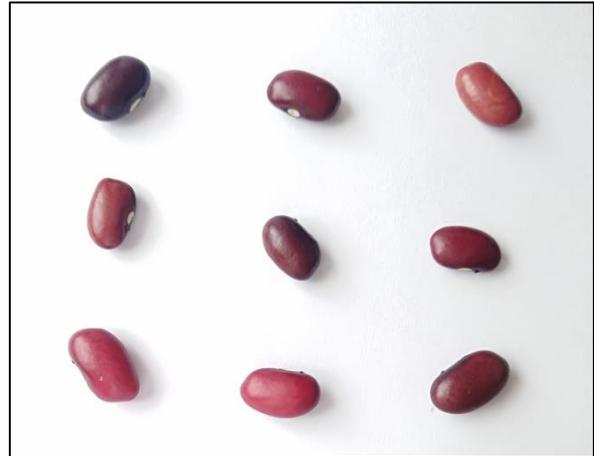
elseif(features(i,1)==6)wrinkle
d=wrinkled+1;
end
end
set(handles.pure, 'String',
pureRKB);set(handles.Food, 'String',
foodB);set(handles.Foreign, 'String',
Foreign);set(handles.Discolored, 'String',
Discolored);set(handles.Broken, 'String',
Broken);set(handles.Wrinkled, 'String',
wrinkled);set(handles.pushbutton5,'Enable','on
');
guidata(hObject,handles);

```

Appendices B - Sample Red Kidney Bean (RKB) Images used for Classification and Grading



“A”



“B”

Figure B-1: “A” Defects RKB classes with Insect board, “B” Healthy RKB classes



“C”



“D”

Figure B-2: “C” Foreign objects RKB classes, “D” Contrasts RKB classes



“E”



“F”

Figure B-3: “E” Defect RKB classes with foreign object, “F” Defect RKB classes (shriveled)