VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

2020-2021

A

Phase 2 Project Report

on

"Grading of Harvested Mangoes based on Quality Evaluation and Maturity Prediction using Machine Learning Technique"

Submitted in the partial fulfillment of the requirement for the VIII Semester Project Phase 2 – 17ECP85 for the award of degree of

Bachelor of Engineering

in

Electronics and Communication Engineering

by

MOHAMMED SHOAIB	1GV17EC018
MOHAN BABU M	1GV17EC019
PAWAN	1GV17EC022
SAJJAD AHMED A	1GV17EC036

Carried out at

Dr. T. THIMMAIAH INSTITUTE OF TECHNOLOGY

Under the guidance of

Ms. Mohana C, M. Tech,

Asst. Professor,

Department of Electronics and Communication.

Dr. T. Thimmaiah Institute of Technology

Oorgaum Post, K.G.F-563120.

(Approved by AICTE, New Delhi, Affiliated to VTU- Belagavi, Approved by Govt. of Karnataka and ISO 21001-2018 certified)

Dr. T. Thimmaiah Institute of Technology

Oorgaum Post K.G.F – 563120
(Approved by AICTE, New Delhi, Affiliated to VTU- Belagavi,
Approved by Govt. of Karnataka and ISO 21001-2018 certified)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING.

CERTIFICATE

Certified that the **Project Work** entitled ***"Grading of Harvested Mangoes Based on Quality Evaluation and Maturity Prediction using Machine Learning Technique"*** is a bonafied work carried out **by Mohammed Shoaib. – 1GV17EC018, Mohan Babu M. - 1GV17EC019, Pawan. - 1GV17EC022, and Sajjad Ahmed A. - 1GV17EC036** in the partial fulfilment for the award of degree of Bachelor of Engineering in **Electronics and Communication Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for the assessment have been incorporated in the phase 2 report deposited in the departmental library. The Phase 2 Project report has been approved as it satisfies the academic requirement in respect of Project Work- 17ECP85 prescribed for the Bachelor of Engineering Degree.

Mohana. C 02/08/2021	Vijaya Bharathi 3/8/2021	28/09/2021
....	....	....
Signature of Guide	**Signature of HOD**	**Signature of Principal**
Ms. Mohana C	**Prof. Vijaya Bharathi M**	**Dr. Syed Ariff**

Name of Examiner	Signature with Date
1. Dr. (Bhuvaneshdhiran T)	1.
2. Rajesh Kumar Kaushal	2.

CONTENT SHEET

LIST OF FIGURES AND TABLES

TABLES

SYNOPSIS

It is very important to do proper Grading of fruits to increase the profit of Agriculture. Nowadays sorting of the fruits like Mango, Banana, Dates and Grapes is performed manually, for this getting adequate manual expert during the period is difficult. This process is time and money consuming also face problem like inconsistency, inaccuracy, inefficiency, lack of objectivity and it is labor intensive process. By using Machine Vision Technique, Several Image Processing Techniques are applied to collect features which are sensitive to Maturity and Quality. For Maturity prediction SVR (Support Vector Regression) technique, for estimation of Quality MADM (Multi Attribute Decision Making) is adopted. Finally, Fuzzy Incremental Learning Algorithm has been used for Grading of Mangoes.

In this system mangoes are graded in three types like Green Mango, Yellow Mango and Red Mango which are based on machine learning method. This system considers RGB values, size and shape of mangoes. Posterior analysis is used to obtain good probability. This helps to train system to detect appropriate maturity of mangoes. This experiment is conducted based on Naive Bayes to compare the performance of both based on accuracy and defective pixels. From the previous system, this system gives the more accuracy as posterior analysis is used. Here we are going to use MATLAB Tool version R2018a.

ACKNOWLEDGEMENT

It is with deep feeling of gratitude we would like to express our sincere thanks to our institution **Dr. T. THIMMAIAH INSTITUTE OF TECHNOLOGY, K.G.F** for providing excellent infrastructure for the completion of the project work.

We wish to express a wholehearted thanks to our **Principal Dr. Syed Ariff** for providing good infrastructure for undertaking this project in college.

We would like to extend hearty thanks to our **Dean (Academics) Prof. Ruckmani Divakaran,** for being constant support of encouragement to carry out the project.

We would like to extend hearty thanks to our **HOD Prof. Vijaya Bharathi M**, for her timely support in the completion of this project work.

We would like to thank our guide **Ms. Mohana C, Assistant Professor**, for her timely support in the completion of this project.

We would like to thank project coordinator **Dr. Jenitha A, Associate Professor**, for her timely support in the completion of this project work.

We would like to thank all teaching and non-teaching staff who were directly and indirectly supported for carrying out this project work.

We extend our hearty thanks to our parents, friends for all the moral support provided during the preparation for the project work.

MOHAMMED SHOAIB **1GV17EC018**

MOHAN BABU M **1GV17EC019**

PAWAN **1GV17EC022**

SAJJAD AHMED A **1GV17EC036**

Chapter 1

INTRODUCTION

Mango is a seasonal and commercial fruit and it is “King of Fruit” in India it is popular due to its taste aromatic fragrance and color. Mangoes are cultivated in different countries like India, Malaysia, Indonesia, Thailand and Sri Lanka. And it is transported in to different countries. Now a day’s sorting of fruits like mango, banana, dates and grapes perform manually, so this purpose getting adequate manual experts during the period is difficult and also this process is time consuming and it faces problems like inconsistency and inaccuracy in judgment as different people are working. The supplies and distributor demand the fruits high quality and good packing. In this project we are using Machine Learning Technique, the system makes able to categorize the mango as over matured, matured and normal. Grading is based on the quality of fruit by considering the attributes value like shape, size and color. In this system the original RGB images are converted into the gray scale for purpose of removing the noise of images [1].

1.1 History of the project

Agriculture is one of the Largest economic sectors and it plays the major role in economic development of our country. Due to this, criteria losses are involved in processing and increasing the demand of fruits of high quality with appearance, there is a need for the development of accurate, fast and focused-quality feature of fruits. Fruit market is getting highly selective, requiring their suppliers to distribute the fruits of high standard of quality and presentation as well. As there is increase in supply of quality fruits within a short period of time has given rise to the development of Automated Grading of Fruits to improve the quality. In general, mango grading is done by using the human experts. Human experts grade the mangoes manually which cause lack of objectivity, efficiency and accuracy. Also, in relation to the evolution of the change of surface color, size, shape, weight may vary according to variety of mango. The appropriate degree of maturity corresponding to the varietal characteristics is also required to withstand transport and handling, and to arrive in satisfactory condition at the pace of destination [8].

In existing system several image processing techniques are applied to collect features which are sensitive to Maturity and Quality. For maturity prediction in terms of actual days to rot Support Vector Regression has been employed and for quality Multi Attribute Decision Making (MADM) system have been adopted. Before experimental works, each mango was presented to five-independent experts for recording experts opinion [8].

The average nearest integer of the five expert's opinion was considered as expert predicted days to rot. Only those mangoes were collected which having days to rot in the range of 5-14 days, and for each case (i.e., days to rot) total 10 number of mangoes were collected, having accumulated total of 100. The same procedure was followed to collect another 100 number of mangoes in second phase. After collection, mangoes were tagged by some serial number and images of mangoes were collected from CCD-camera starting from day-0. Then the mangoes were placed in the packages in the same manner as used for the transport of the mangoes to different cities. On the next day each mango was taken out from the package carefully for taking image for the day-1 and also presented to the 5-experts and their predicted days to rot were recorded, the same process was followed for all days [8].

1.2 Reason for selecting the project

The main aim of this project is to find out the mango's maturity and quality in the form of size, shape and surface defects of mangoes images. In this project we are using Machine Learning Technique which is easy to use and no human intervention is needed (automation) which will help to increase the profit of vendors by reducing the time consumption and increase in the accuracy level. Here Naive Bayes Algorithm is applied which uses a similar method to predict the probability of different class based on various attributes [1].

Posterior analysis is used to obtain the good probability. In the present system they are using several image processing techniques for Maturity and Quality prediction, they employed machine vision technique which leads to less in accuracy and time consumption is high [1].

1.3 Requirements

i) Computer (Operating System: Windows 7/10 with 64bit, 4GB RAM, 2Ghz processor)
ii) MATLAB R2018a
iii) Datasets (kaggle.com [18], github.com [19, 20, 21], data.mendeley.com, kent.com) [Training Images = 100; Testing Images = 50] for each colour
iv) Google Drive

Chapter 2

LITERATURE SURVEY

A literature survey or literature review in a project is a type of review articles. It includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and do not report new or original experimental work. It is a basis for research in nearly every academic field. Concentrate on the own field of expertise.

2.1 Details of Literature

Dnyaneshwari Pise, G. D. Upadhye [1] proposed "Grading of Harvested Mangoes Quality and Maturity based on Machine Learning Techniques", It is very important to do proper grading of fruits to increase the profit of agriculture and food industries. Process of grading of mangoes generally based on the physical properties of mangoes. In this paper, the original RGB images are converted into grayscale for the purpose of removing the noise of images. After that grayscale images are converted into binary images for analyzing the surface area of mangoes. Naive Bayes and SVM (Support Vector Machine) with Posterior Analysis technique is adopted to grade the mangoes according to size, shape and color of images. Posterior Analysis is used to obtain good probability. This helps to train a system to detect appropriate maturity of mangoes. The system gives the more accuracy as posterior analysis is used.

Dah-Jye Lee, James K. Archibald and Guangming Xiong [2] proposed "Rapid Color Grading for Fruit Quality Evaluation using Direct Color Mapping". In this paper, effective and user-friendly color mapping concept for automated color grading that is well suited for commercial production. The color mapping method uses preselected color of interest specific to a given application to calculate a unique set of co-efficient for color space conversion. Here new color mapping concept of converting 3-D color spaces to 1-D color indices for automated color grading. The main approach is to convert 3-D RGB values into a simple 1-D color space. The three-dimensional RGB color space is converted into a small set of color indices unique to the application. The proposed method makes it easy for a human operator to specify and adjust color preference settings Tomato and Date maturity evaluation and Date surface detection are used to demonstrate the performance of color mapping concept.

Chandra Sekhar Nandi, Bipan Tude and Chiranjib Koley [3] proposed "Machine Vision based Automatic Fruit Grading System using Fuzzy algorithm". This paper proposes a machine vision-based scheme for automatic grading of fruits according to their maturity level and quality. The manual grading by visual inspection poses problems in maintaining consistency and accuracy; this is also time consuming and labor-intensive process. In this project a new prototype computer vision based automatic fruit grading system is proposed. The automated system collects video image from the CCD camera placed on the top of a conveyer belt carrying mangoes, then it processes the images in order to collect features which are sensitive to the maturity level and quality. Finally, fuzzy rule-based algorithm is used to sort the fruits into four grades.

Sapan Naik, Bankim Patel and Rashmi Pandey proposed [4] "Shape, Size and Maturity features Extraction with Fuzzy Classifier for Non-Destructive Mango Grading", In this paper three features namely shape, size and maturity are used for mango fruit grading. Here the proposed methodology is divided in three phases: In first phase mangoes are classified either well-formed or deformed using eccentricity, extent and cross ratio properties of shape. Second phase discuss about size and maturity classification. In this phase mangoes are classified into small, medium or big size and unripe, partially ripe or ripe maturity. At final phase, decision making theory is used to grade mango in class I, class II and class III. In this proposed model fuzzy system is used for mango grading based on maturity and size. Gaussian mixture model is applied to judge the ripeness of mango, which uses 15 main and 12 derived features in RGB color space. Size is estimated using area calculation. By using all these technologies, we have obtained 96% accuracy for size feature, 92% accuracy for shape feature and 93.5% accuracy for maturity. In grading using the combination of all these three features, we have achieved 90% of accuracy.

Yogitha. S and Sakthivel. P [5] proposed "A Distributed Computer Vision System for Automated Inspection and Grading of Fruits". A computer machine vision system that can be used for automatic high-speed fruit sorting and grading is proposed. The grading and sorting of the fruits is based on external parameters namely color, size and shape. The proposed scheme for a real time machine vision inspection system based on diameter and weight fruit sorter makes use of distributed network architecture to interface the field IO devices and camera inspection system to the computer system through Gigi LAN environment in a flexible way. This project is planned to do in visual studio using OpenCV. The process involved for estimating the color information and geometry parameters makes use of sequence of complex library functionality like removing the noise, detecting the edges, smoothing, dilate, erode, filling, deblurring, filtering, histogram, color values, pixel averaging etc.

Kedar Patil, Shriniwas Kadam, Suraj Kale, Yogesh Rachetti, Kiran Jagtap and Dr. K.H. Inamdar [6] proposed "Machine Vision Based Autonomous Fruit Inspection and Sorting". The main aim of this paper is to design and development of a low-cost automated mechatronic system for inspection and grade-wise sorting of fruits and vegetable. Fruits are sorted as immature, rotten and edible. Edible fruits can be further graded and sorted according to size. This paper explains the technique to collect data like size, color, defects of fruits and vegetables and it is sort out in different categories. Suitable MATLAB Algorithms like Conversion to Binary Image, Area calculation and Average pixel value calculation are used. For sorting purpose Arduino-Uno-Microcontroller is used. The algorithm used in this paper runs successfully and can be modified at any point of time depending upon lighting conditions varying from place to place and time to time. Also, other types of fruits can be sorted with little modifications in algorithm. By using longer lengths of belt to the proposed system accuracy can be increased in terms of speed.

Ms. Seema Banot, Dr. P.M. Mahajan [7] proposed "A Fruit Detecting and Grading System Based on Image Processing". Different types of algorithms are used to extract feature of the fruit image. With the help of this feature fruit is detected and graded according to quality of fruit. This can be done by using different type of classifier. The grading system have the advantage of high accuracy, high speed, and low cost. This work aimed to study different types of algorithms used for quality grading to develop an algorithm for detecting and sorting of the fruit from the image. The morphological feature is used and color can be extracted which is further more used to identify class of the fruit using Neural Network. In this paper various image processing techniques are used such as fuzzy logic technique – This method has been implemented using MATLAB tool and is suitable for various fuzzy environments, main advantages of this method are use of fuzzy-interference engine without depending on human expert. The accuracy achieved was 80%, Artificial Neural Network Technique – This technique is used to develop algorithms and they can be successfully detect and classify the tested disease and get better result for color and morphology and they reported 90% of the result, K-Nearest Neighbor Classifier (KNN) – This technique is used to identify the input data by comparing it with the trained data. KNN Classifier is also used to classify fruit on the basis of RGB color value, shape roundness value, area and perimeter value of the fruit and they reported 90% accuracy.

Chandra Sekhar Nandi, Bipan Tudu and Chiranjib Koley [8] proposed “A Machine Vision Technique for Grading of Harvested Mangoes based on Maturity and Quality”. The proposed scheme works on Machine Vision based Technique for Grading of Mangoes in four different categories. In this system video image is captured by CCD (Charge Couple Device) camera place on the top of conveyor belt carrying mangoes. For maturity prediction in terms of actual-days-to-rot Support Vector Regression (SVR) has been employed and For estimation of quality-attribute Multi-Attribute Decision Making (MADM) system have been adopted. Fuzzy-Incremental Algorithm has been used for grading based on maturity and quality. The proposed system not only predicts maturity level and quality level but also predicts actual-day-to-rot of mangoes and also it can automatically grade the entire lot of mangoes according to vendor specific inputs on quality and transportation delay. The performance accuracy achieved using this proposed system for grading of mango fruit is nearly 87%.

Mr. Sumit S Telang, Prof. S.M. Shirsath [9] proposed “Fruit Quality Management Using Image Processing”. This system determines the quality of a fruit by its color, size and weight. The sorting process depends on computing the image of fruit and analyzing the image using image processing technique to discard defected fruits. The main emphasis to do the quality check with a short span of time, so that maximum number of fruits can be scrutinized for quality in minimum amount of time. The proposed automated classification and grading system is designed to combine three processes such as feature extraction, sorting according to color and a grading according to size. The entire system is designed over a MATLAB Software to inspect the color and size of the fruit. The main objective of this paper is detection of good or bad quality fruits efficiently. This detection is achieved by processing the image of fruit. Algorithms are developed for checking different parameters which are responsible for quality of a fruits. The results obtained can be used for statistical analysis which further decreases detection time.

Dr. S. Usha, Dr. M. Karthik, R. Jenifer [10] proposed “Automated Sorting and Grading of Vegetables using Image Processing”. The computer vision-based system for automatic and sorting of the agricultural products like Strawberry and Brinjal based on maturity level is presented in this paper. In the proposed system pre-processing, color detection method and segmentation process are applied over the image for classification. The purpose of pre-processing technique is to remove the noise and enhance the image quality. The system is implemented by K-means clustering segmentation and color detection process with strawberry and brinjal feature extraction for various features like entropy, mean and standard deviation is calculated. Color Detection model is used to identify the defected part with threshold level then the threshold value for the strawberry is 98 and threshold value for brinjal is 70. Then the segmentation technique is K-means clustering segmentation concept. This concept split into the different cluster like affected part, non-affected part of strawberry and a brinjal. After segmentation process feature extraction can be done by the different maturity level of strawberry and brinjal by using various features like entropy, standard deviation and mean. The various maturity levels are Ripe-Fruit, Unripe-Fruit, Affected Fruit. Based upon feature values classify the strawberry and brinjal in a different maturity level. The whole process is done by simulation in MATLAB and Desktop Application was developed by using MATLAB Guide.

Shwetha S Deulkar, Dr. Sunita. S. Barve [11] proposed "Feature based fruit Quality Grading System using Support Vector Machine (SVM)". In this paper, they have developed computer vision based on image acquisition and classification system for a fruit grading. This system is divided into three modules Image Pre-Processing, Image Segmentation and Classifier. In the first module, we are extracting the image features such as image quality, area, perimeter, mean, variance, color, intensity. Next OTSU Algorithm work on automatic threshold calculation and image binarization part. In the second module K-means clustering is used for image segmentation. In last module the vectors of color, intensity, segments and image quality feature were utilized for training of Support Vector Machine (SVM) structure. The image K-means clustering approach is embedded with Support Vector Machine (SVM) classifier to improve the speed and accuracy of the classifier. The automated grading and sorting system should be developed to replace traditional visual inspection system. OTSU thresholding algorithm is used for the automatic threshold calculation and conversion of gray color image to bicolor image. Algorithm considers that the image consists of only two-color set of pixels i.e., Object and Background. The purpose of Support Vector Machine (SVM) is to classify the image that has similar features into one group. Support Vector Machine (SVM) is categorized into two-classes i.e., Linear and Non-Linear. The modelling goal of the Support Vector Machine (SVM) is to find an optimal hyper-plane which divides the dataset such that the margin between dataset is maximized Support Vector Machine (SVM) classification results are more accurate compare to any other classifier. The proposed tomato grading system is very effective in terms of accuracy, efficiency, time and cost.

2.2 Objective of the project

To achieve a higher Efficiency and Accuracy in detecting and grading of mangoes based on maturity and Quality, which will reduce the manual work in grading of Mangoes and reduces the time consumption.

2.3 Problem Statement

Several Image Processing Techniques are applied based on Maturity and Quality, for Maturity prediction SVR (Support Vector Regression), for estimation of Quality MADM (Multi Attribute Decision Making) techniques are employed. Accuracy is less and time consumption is more.

Chapter 3

METHODOLOGY

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques [17].

Methodology refers to the overarching strategy and rationale of project. It involves studying the methods used in the field and the theories or principles behind them, in order to develop an approach that matches an objective.

3.1 Introduction

A methodology does not set out to provide solutions it is therefore, not the same as a method. Instead, a methodology offers the theoretical underpinning for understanding which method, set of methods, or best practices can be applied to a specific case, for example, to calculate a specific result. It has been defined as follows:

1. "The analysis of the principles of methods, rules, and postulates employed by a discipline";
2. "The systematic study of methods that are, can be, or have been applied within a discipline";
3. "The study or description of methods".

The Methodology is the general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it. These methods, described in the methodology, define the means or modes of data collection or, sometimes, how a specific result is to be calculated [17].

Methodology does not define specific methods, even though much attention is given to the nature and kinds of processes to be followed in a particular procedure or to attain an objective. Methodology may be visualized as a spectrum from a predominantly quantitative approach towards a predominantly qualitative approach. Although a methodology may conventionally sit specifically within one of these approaches.

When proper to a study of methodology, such processes constitute a constructive generic framework, and may therefore be broken down into sub-processes, combined, or their sequence changed [17].

A paradigm is similar to a methodology in that it is also a constructive framework. In theoretical work, the development of paradigms satisfies most or all of the criteria for methodology. An algorithm, like a paradigm, is also a type of constructive framework, meaning that the construction is a logical, rather than a physical, array of connected elements.

Any description of a means of calculation of a specific result is always a description of a method and never a description of a methodology. It is thus important to avoid using methodology as a synonym for method or body of methods. Doing this shifts it away from its true epistemological meaning and reduces it to being the procedure itself, or the set of tools, or the instruments that should have been its outcome. A methodology is the design process for carrying out research or the development of a procedure and is not in itself an instrument, or method, or procedure for doing things [17].

Methodology and method are not interchangeable. In recent years, however, there has been a tendency to use methodology as a "pretentious substitute for the word method". Using methodology as a synonym for method or set of methods leads to confusion and misinterpretation and undermines the proper analysis that should go into designing research.

Methodology comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge such that the methodologies employed from differing disciplines vary depending on their historical development. This creates a continuum of methodologies that stretch across competing understandings of how knowledge and reality are best understood. This situates methodologies within overarching philosophies and approaches [17].

Essentially, a methodology is a collection of methods, practices, processes, techniques, procedures and rules. In project management, methodologies are specific, strict and usually contain a series of steps and activities for each phase of the project’s life cycle [17].

3.2 Block Diagram of Grading of Mangoes

The proposed Mango Grading System determines Mangoes Maturity and Mangoes Quality in the form of Size, Shape and Surface Defects of Mangoes Images. By using various techniques increases the profit of vendors and time consumption is less.

The Block Diagram of Grading of Mangoes as shown in the Figure 3.1, it mainly consists of the input image, image pre-processing techniques, training of an image, Naive Bayes and Posterior analysis [1].

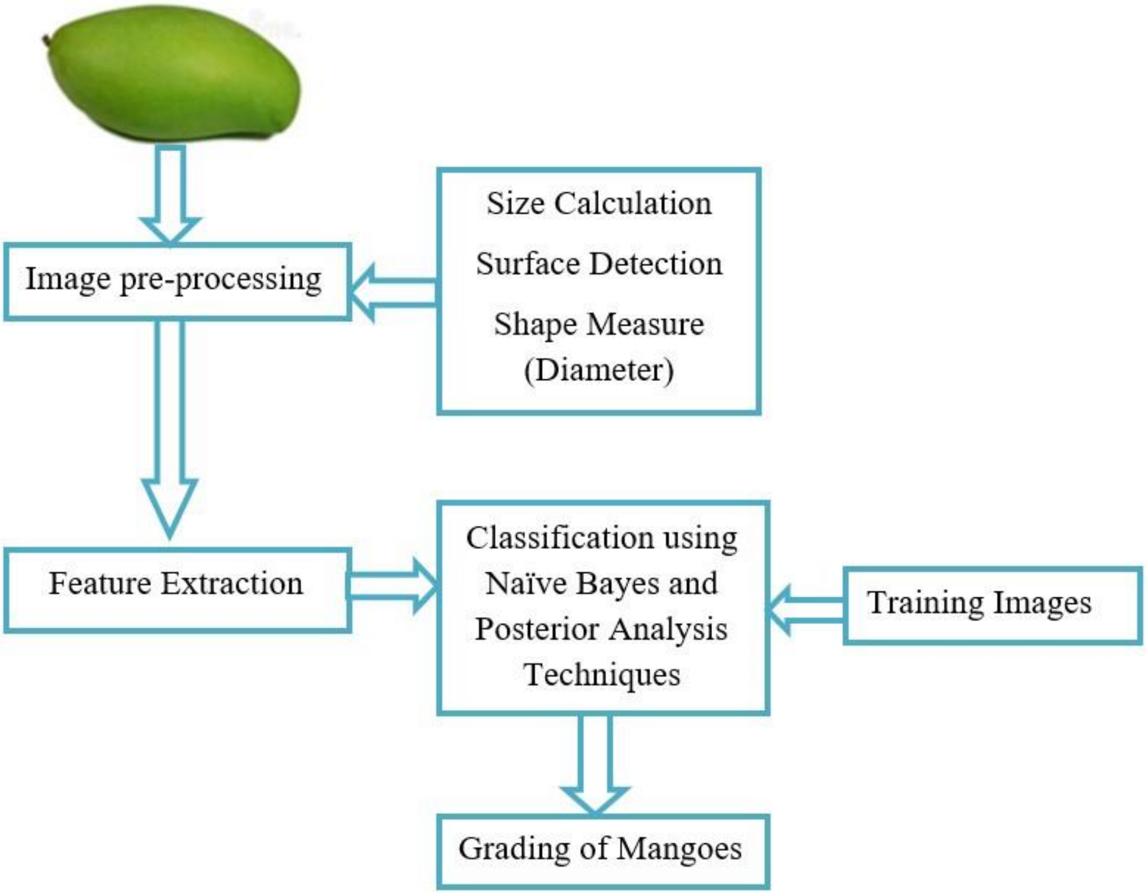

Figure 3.1: Block Diagram of Grading of Mangoes

Image pre-processing techniques are used to calculate the size of the Mango, surface of the mango is detected and shape is measured. The classification is done by adopting Naive Bayes and Posterior Analysis techniques. Posterior analysis is used to obtain good probability. These techniques will train the images of Mangoes to predict the quality, maturity and color of the mango and the output obtained is graded Mango.

3.2.1 System Architecture

The proposed Mango Grading System determines Mangoes Maturity and Mangoes Quality in the form of size, shape and surface defects of Mangoes Images. Using these techniques increases the profit of vendors and time consumption is less [3].

The proposed system working is based on the different type of mangoes images and which are different type of maturity level:

Input: - Image of Mango.

Output: - Grading of mangoes according to maturity level.

The input mango image is pre-processed to remove the noises and improve the quality of the image. [12].

Process of grading of mangoes is done generally based on the physical properties of mangoes. The original RGB images are converted into the gray scale for purpose of removing the noise of images. After that gray scale images are converted into binary images for the analyzing surface area of mangoes. Naive Bayes with Posterior Analysis method is used to grade the mangoes according to the size, shape and color of Mango images [3].

The aim is to grade the mangoes automatically with help of the image processing algorithm and Naive Bayes classifier for the classification. The traders of the mangoes can use this system to grade the mangoes and separate them according to the quality automatically. In industry, time and efficiency are the major factors that can't be achieved by the human inspection system because it is difficult for the human to inspect large number of mangoes [12].

3.2.2 Pre-Processing

Image noise is defined as distinct pixels which are not similar in appearance with the neighborhood pixels. Over segmentation occurs mainly due to presence of the noise and unimportant fluctuation which produces non real minima. Main objective of the pre-processing stage is to smooth the original image by removing the noise effect and enhance the image quality of the mangoes.

Image processing is done for manipulation of image data and the task which can influence is distributed into categories of three i.e., understanding, analysis, and processing done on image.

The input image is pre-processed using the different methods as Gray Scale, Binary Image Representation and Maximum height. Features of Mangoes are considered as size, surface and shape. These features are calculated by calculating RGB values of images, size and diameter of image. Size of images are calculated using edge detection method [1].

Figure 3.2 RGB Image to Grayscale Image

A Grayscale images, in which it carries only intensity of the images. Grayscale image is also known as black and white image as shown in Figure 3.4, black at the weakest intensity as compared to the white [1].

Binary images are created from threshold value of a grayscale or color image for separating an object as shown in Figure 3.5. The color of Mango images is white which is referring to the foreground color and background color of images are black [1].

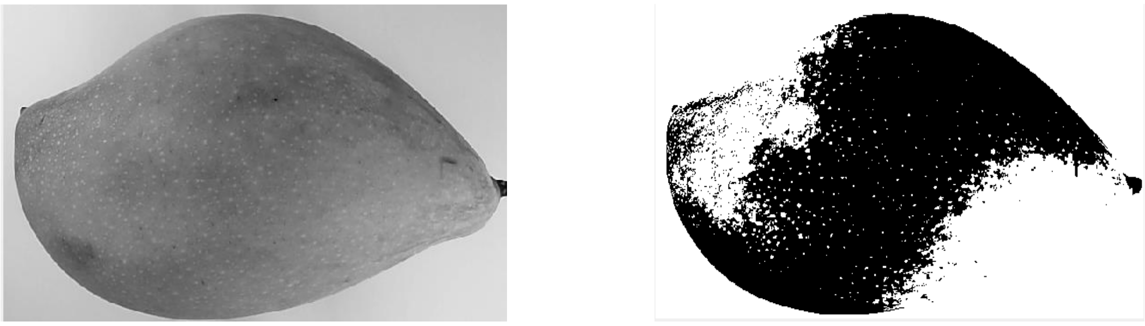

Figure 3.3 Grayscale Image to Binary Image

3.2.3 Feature Extraction

Features are extracted to comprise the appropriate information from the data which has been input and then the desired task performs with the help of input data which are reduced in its place of comprehensive original data. Extraction of features is used for reducing the resource in the suitable amount from the set of large data. Image processing is the very important area of the feature extraction. In which algorithm are used to detect various desire portion of the image. Feature extraction is done so that the feature from the object can be extracted. Features are extracted with respect to various aspects namely, Geometrical extraction of feature, texture extraction of feature, color extraction of feature, and histogram extraction of feature [12].

The feature extraction is done to measure the maturity level and affected part of Mango. Feature extraction is a method of capturing visual content of an image. The objective of feature extraction process is to represent raw image in its reduced form to facilitate decision making process such as pattern classification. Entropy, Mean and Standard deviation used to extract gradient feature in proposed project. A feature is extracted in order to allow a classifier to distinguish between diseased part and riped fruit [10].

Image Entropy formula is expressed as,

$$\text{Entropy} = - \sum_i P_i \ \log_2 Pi \qquad \text{……………..……. (3.1)}$$

Where,

i) P_i is the probability that the difference between 2 adjacent pixels is equal to i.

ii) Log_2 is the base 2 logarithm

The formula for the sample standard deviation is

$$S = \sqrt{\frac{\sum_{i=1}^{N}(x-\bar{x})^2}{N-1}} \quad \text{………………..……. (3.2)}$$

Where,

i) N is number of Pixel points

ii) $\bar{x}$ is the mean of x_i

iii) x_i is each values of pixels

The mean is the average of all numbers and is sometimes called the arithmetic mean.

The mean formula is expressed as,

$$\bar{X} = \sum \frac{X}{N} \quad \text{……………………. (3.3)}$$

Where,

i) $\bar{X}$ is the mean

ii) N is the number of values

3.2.4 Techniques Used

Naive Bayes for classification (With training data) and Posterior Analysis techniques:

1. Naive Bayes

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature [16].

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’ [16].

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods [16].

Bayes theorem provides a way of calculating posterior probability P(c/x) from P(c), P(x) and P(x/c). Look at the equation below:

$$P(c/x) = [\ P(x/c)\ P(c)\ ] / P(x) \qquad \text{(3.4)}$$

$$P(c/x) = P(x_1/c) * P(x_2/c) * \ldots * P(x_n/c) * P(c) \qquad \text{(3.5)}$$

Where,

i) P(c/x) is the posterior probability of class (c, target) given predictor (x, attributes).
ii) P(c) is the prior probability of class.
iii) P(x/c) is the likelihood which is the probability of predictor given class.
iv) P(x) is the prior probability of predictor.

2. Posterior Analysis

Naive Bayes (NB) is a very simple algorithm. This can be seen in steps to resolve the posterior probability calculation [1]. Components in NB are as follows this point below:

Posterior = Likelihood * Prior Evidence ……………….…. (3.6)

Following steps are involved to accomplish classification:

I. Prior Probability:

Prior probability as value for a certain class is calculated from experience of previous events. The more data the past is used, then the values of the prior probability, the better and sometimes the prior probability value is used to determine the classification result when some prior probability values between classes [1].

Examples:

P(Class-1) P(Class-2) .... P(Class-n) …………………….. (3.7)

Where n specifies the number of classes and i = 1, 2, 3 ... n.

II. Conditional Probability:

Conditional probability is a probability values that has a requirement previous events and events into the conditions are a particular class, such as the i^{th} class "Class-i" meaning a requirement that the incident is an event that happened occurs first, followed by the emergence of other events, such as the incident of "Criteria-1", "Criteria-m", where m denote the number of feature or criteria or parameter [1]. So, write conditional probability variable i.e.,

P(Criteria-1|Class-i) * P(Criteria-2|Class-1) * P(Criteria-m|Class-i) ………. (3.8)

III. Marginal Probability (Evidence):

Evidence is the probability of single occurrence of one or a set of features Evidence "P(Criteria-1, Criteria-2, Criteria-m)" can be calculated by summing the result of "likelihood * prior" that run on all existing classes [1].

IV. Posterior Probability:

Posterior probabilities have almost similar characteristics with conditional probability and conditional probability is the opposite of the sequence of events from the point of view [1].

V. Grading of Image:

Finally, after applying all logic and algorithms graded image of mangoes is obtained.

Chapter 4

MACHINE LEARNING ALGORITHM

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly [22].

4.1 Types of Machine Learning Algorithm

Machine Learning Algorithms are broadly classified into three types: -

1. **Supervised Learning**
   This algorithm consists of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, generates a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Naive Bayes Algorithm Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.
2. **Unsupervised Learning**
   In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means.
3. **Reinforcement Learning**
   Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process.

4.2 Types of Supervised Learning

There are six types of Supervised Learning Algorithm: -

1. **Regression**

   In regression, a single output value is produced using training data. This value is a probabilistic interpretation, which is ascertained after considering the strength of correlation among the input variables. For example, regression can help predict the price of a house based on its locality, size, etc.

2. **Classification**

   It involves grouping the data into classes. When the supervised learning algorithm labels input data into two distinct classes, it is called binary classification. Multiple classifications mean categorizing data into more than two classes.

3. **Naive Bayesian Model**

   Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

   Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

4. **Random Forest Model**

   The random forest model is an ensemble method. It operates by constructing a multitude of decision trees and outputs a classification of the individual trees. For Example: To predict which undergraduate students will perform well in GMAT – a test taken for admission into graduate management programs. A random forest model would accomplish the task, given the demographic and educational factors of a set of students who have previously taken the test.

5. **Neural Networks**

   This algorithm is designed to cluster raw input, recognize patterns, or interpret sensory data. Despite their multiple advantages, neural networks require significant computational resources. It can get complicated to fit a neural network when there are thousands of observations. It is also called the 'black-box' algorithm as interpreting the logic behind their predictions can be challenging.

6. **Support Vector Machine (SVM)**

   Support Vector Machine (SVM) is a supervised learning algorithm developed in the year 1990. It draws from the statistical learning theory developed by Vap Nick.

   SVM separates hyperplanes, which makes it a discriminative classifier. The output is produced in the form of an optimal hyperplane that categorizes new examples. SVMs are closely connected to the kernel framework and used in diverse fields. Some examples include bioinformatics, pattern recognition, and multimedia information retrieval.

4.3 Naive Bayes Algorithm

Naive Bayes is a probabilistic machine learning algorithm that can be used in a wide variety of classification tasks. Typical applications include filtering spam, classifying documents, sentiment prediction etc. The name naive is used because it assumes the features that go into the model is independent of each other. That is changing the value of one feature, does not directly influence or change the value of any of the other features used in the algorithm [23].

It is a subset of Bayesian decision theory. It’s called naive because the formulation makes some naïve assumptions. This can be used to classify text. Classifies may put into human-readable form. It is a popular classification method in addition to conditional independence, overfitting, and Bayesian methods.

It is a reasonable classifier in this sense and has minimal storage and fast training, it is applied to time-storage critical applications, such as automatically classifying web pages into types and spam filtering [23].

4.3.1 Mathematical Model of Naive Bayes Algorithm

RGB values of mangoes images are calculated by using the following equation:

$$I_K = R, G, B = \frac{1}{r}\sum_{N=1}^{R} \sum_{M=1}^{C}(IK * BI) \quad \text{.....…………..(4.1)}$$

Here, BI are the Binary Image of mangoes. IK is the captured RGB image, C and R are the total number of columns and rows of the mangoes image.

Derived Features:

Difference of average values of R, G and B calculate of the mango images are:

$(A_G - A_R)$, $(A_G - A_B)$ and $(A_R - A_B)$ ……..…………(4.2)

Difference of average value R, G and B value for the equator, stalk, apex, region respectively.

Diameter:

Diameter are measuring using this formula:

$$d_g(x, y) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2} \quad \text{……………………} (4.3)$$

Naive Byes Algorithm perform mango classification by using the distance between the feature value of unknown mango with the feature value of stored mango examples after that algorithm will find out the nearest examples to unknown mango.

Accuracy:

The posterior probability is the probability of the parameters θ given the evidence:

X: P(θ|X)

It contrasts with the likelihood function, which is the probability of the evidence given the parameters: P(θ|X)

Let us have a prior belief that the probability distribution function is P(θ) and observations x with the likelihood P(θ|X), then the posterior probability is defined as:

$$P(\theta|X) = \frac{P(\theta|X, P(\theta))}{P(X)} \quad \text{…………………} (4.4)$$

Naive Byes algorithm perform mango classification by using the feature value of mango with the feature value of stored mango examples after that algorithm will find out the nearest examples to mango [1].

Steps in Naive Byes Algorithm are given below:

Input: Training Images, Testing Images of Mangoes.

Output: Graded Mangoes according to maturity level.

Step I: Read the images of mangoes.

Step II: Preprocessing of images for the feature extraction.

Step III: Feature of mangoes images like RGB values and diameter of mangoes are extracted.

Step IV: Feature extraction after that it crate a Frequency Table of dataset.

Step V: Creating Likelihood table according to the Probabilities values: Overcast Probability = X and Probability of outcome = Y.

Step VI: Read the testing images from data set and follow the Step II and IV.

Step VII: Using Naive Bayes equation to evaluate the Posterior Probability for each class. The class with highest posterior probability is the outcome of prediction [1].

Advantages of Naive Bayes Classifier:

- Naive Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems [23].

Chapter 5

SYSTEM IMPLEMENTATION

The proposed system is stimulated using the MATLAB2018a.

5.1 Introduction to MATLAB

MATLAB was developed primarily by Cleve Moler in the 1970's. Derived from FORTRAN subroutines LINPACK and EISPACK, linear and eigenvalue systems. It was developed primarily as an interactive system to access LINPACK and EISPACK. It gained its popularity through word of mouth, because it was not officially distributed. It was rewritten in C in the 1980's with more functionality, which include plotting routines. The MathWorks Inc. was created (1984) to market and continued development of MATLAB [25] .

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects [25].

It is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, It is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research [25].

It has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization [25].

5.2 Introduction to Programming in MATLAB

The commands entered in the Command Window cannot be saved and executed again for several times. Therefore, a different way of executing repeatedly commands with MATLAB is:

1. To create a file with a list of commands
2. Save the file and
3. Run the file.

The corrections or changes can be made to the commands in the file. The files that are used for this purpose are called script files or scripts for short i.e.,

- M-File Scripts
- M-File Functions [25]

5.2.1 M-File Scripts

A script file is an external file that contains a sequence of MATLAB statements. Script files have a filename extension .m and are often called M-files. M-files can be scripts that simply execute a series of MATLAB statements, or they can be functions that can accept arguments and can produce one or more outputs [25].

5.2.2 M-File Functions

Functions are programs (or routines) that accept input arguments and return output arguments. Each M-file function (or function or M-file for short) has its own area of workspace, separated from the MATLAB base workspace [25].

5.2.3 Parts of MATLAB

There are three main parts of MATLAB: -

- **Editor Window:** The MATLAB Editor Window is a simple text editor where you can load, edit and save complete MATLAB programs. The Editor window also has a menu command (Debug/Run) which allows you to submit the program to the command window. Many of the assignments in editor window will require the use of executable m-files. It will save these m-files to a convenient location (perhaps in the document folder, create a new folder named MATLAB and save them in the folder). Once m-file saved to the computer, to launch MATLAB and open the m-file (you can use the “Open” icon on the MATLAB toolbar). The m-file should now will be open in the Editor Window. If you need to edit an m-file, it will be necessary to use the Editor Window.

- **Command Window:** Once you have opened your m-file in the Editor Window, you can now call and execute that m-file while in the Command Window. Directions for using specific m-files are included in the m-file section of this document.
- **Workspace Window:** This window shows the user what data structures are available for use. It also gives the dimensions of these data structures. Users can manipulate these variables when working in the Command Window. For example, plots can be created using the appropriate variables [25].

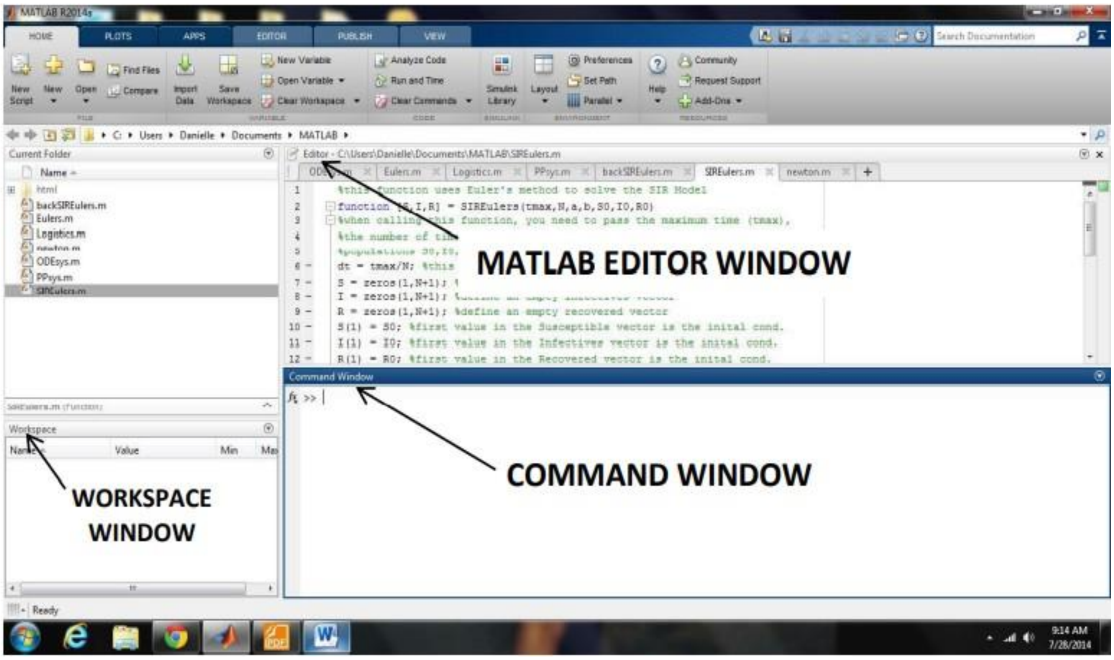

Figure 5.1: Parts of MATLAB

5.2.4 Features of MATLAB

Following are the basic features of MATLAB:

- It is a high-level language for numerical computation, visualization and application development.
- It also provides an interactive environment for iterative exploration, design and problem solving.
- It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
- It provides built-in graphics for visualizing data and tools for creating custom plots.

- MATLAB's programming interface gives development tools for improving code quality, maintainability, and maximizing performance.
- It provides tools for building applications with custom graphical interfaces.
- It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel [25].

5.2.5 Uses of MATLAB

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including:

- Signal processing and Communications
- Image and video Processing
- Control systems
- Test and measurement
- Computational finance
- Computational biology

5.3 System Requirements – MATLAB2018a

The System Requirements of MATLAB2018a are as follows: -

64-Bit MATLAB, Simulink and Polyspace Product Families				
Operating Systems	Processors	Disk Space	RAM	Graphics
Windows 10 Windows 8.1 Windows 7 Service Pack 1 Windows Server 2016 Windows Server 2012 R2 Windows Server 2012	**Minimum** Any Intel or AMD x86-64 processor **Recommended** Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support	**Minimum** 2 GB of HDD space for MATLAB only, 4-6 GB for a typical installation **Recommended** An SSD is recommended A full installation of all MathWorks products may take up to 22 GB of disk space	**Minimum** 4 GB **Recommended** 8 GB For Polyspace, 4 GB per core is recommended	No specific graphics card is required. Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended. GPU acceleration using the Parallel Computing Toolbox requires a CUDA GPU. See *GPU Computing Support* for details.

5.4 System Implementation of Proposed Model

Step 1: Writing code in the Editor Window

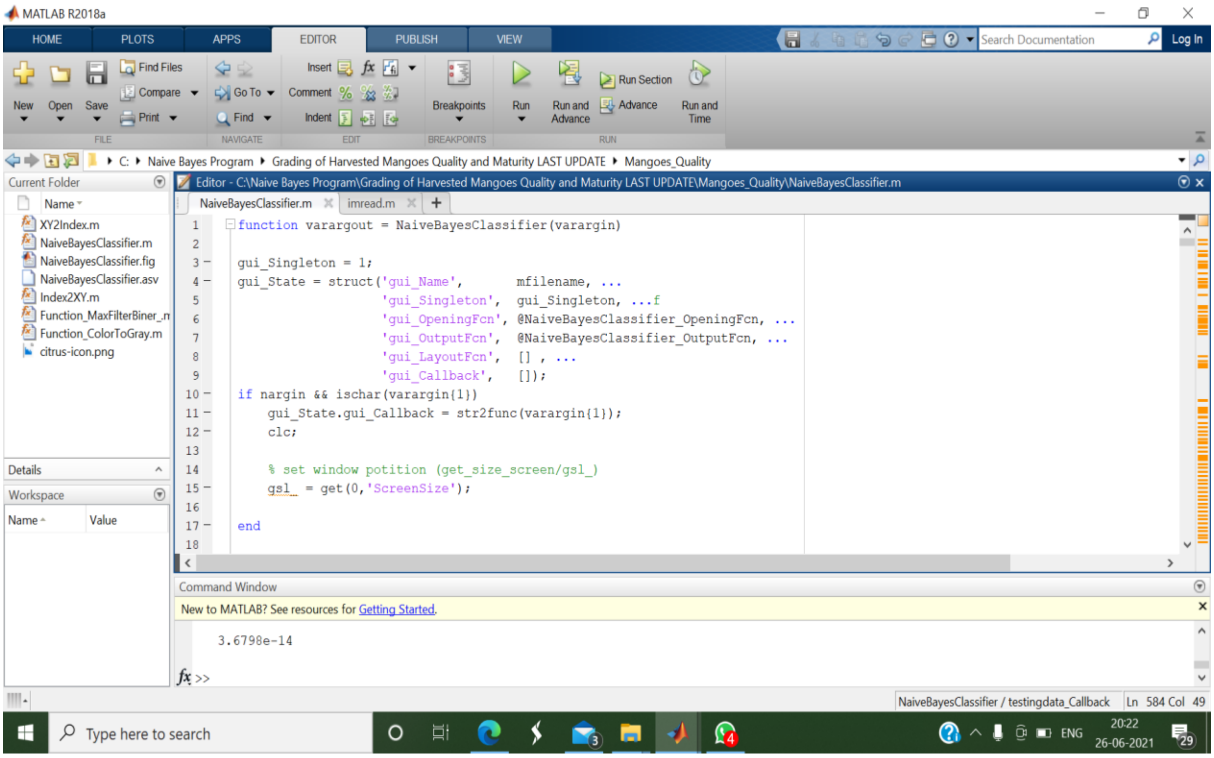

Figure 5.2: Writing code in the Editor Window

Step 2: Importing Data from the C-Drive

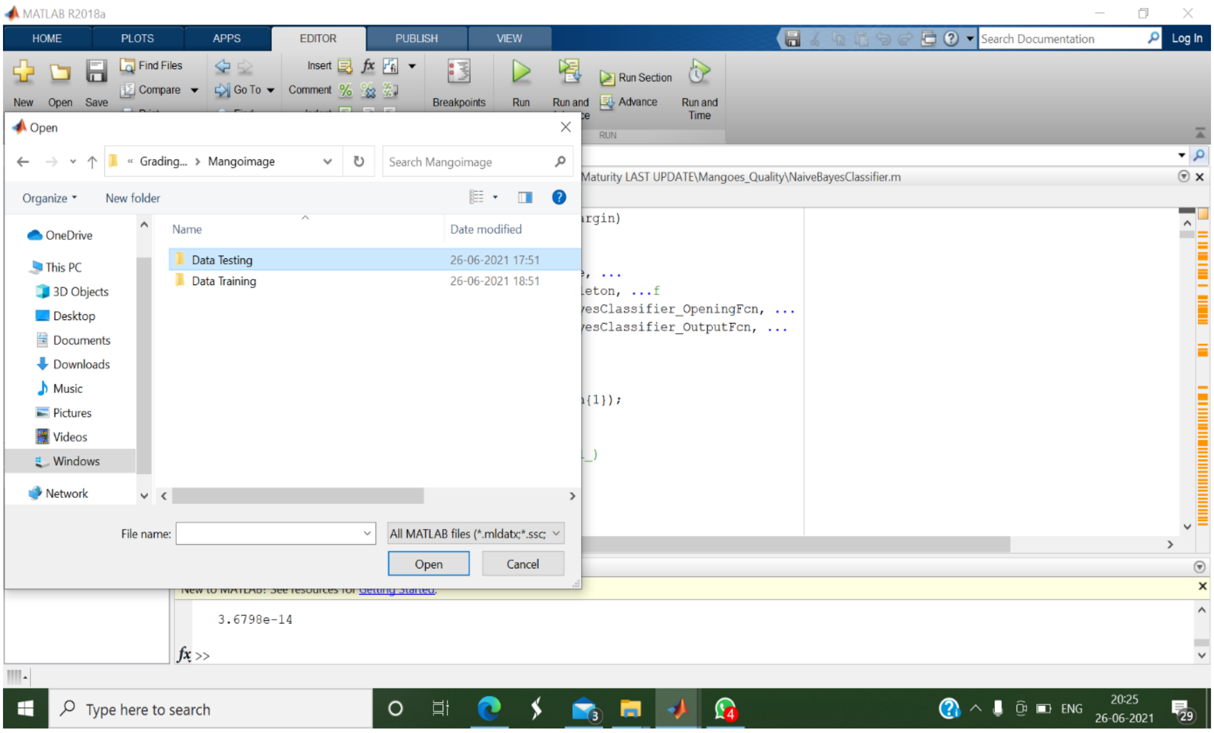

Figure 5.3: Importing Data from the C-Drive

Step 3: Checking for the Outputs

The final output obtained in MATLAB for three different color of mango images is as shown below:-

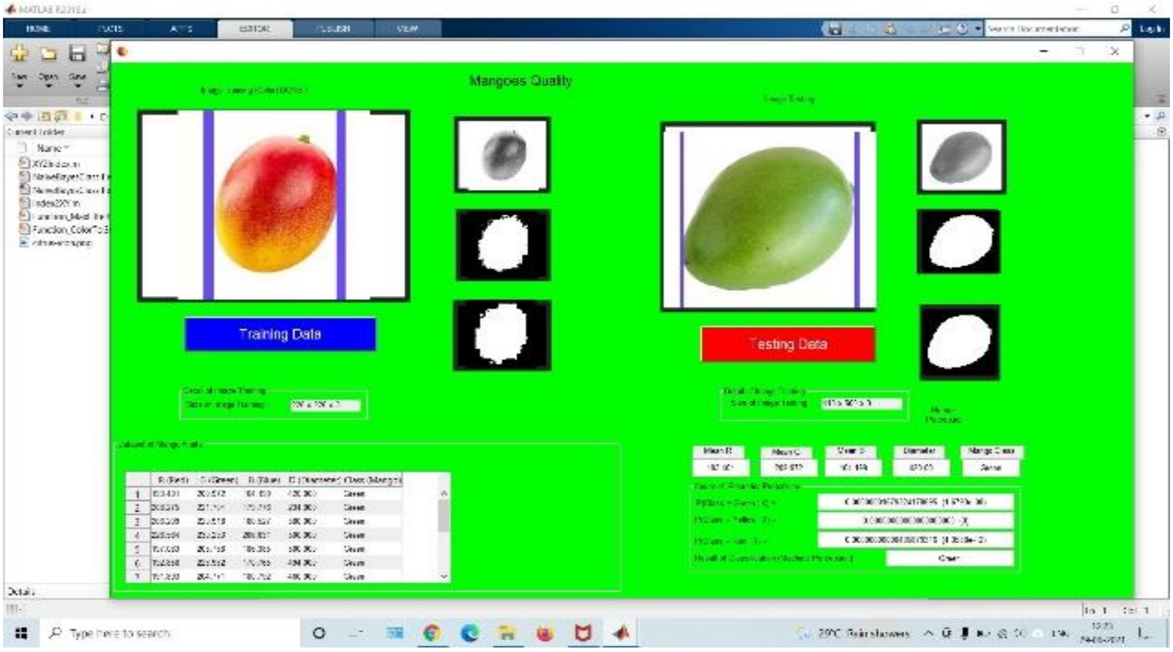

Figure 5.4: Output for Green Mango Image in MATLAB

Figure 5.5: Output for Yellow Mango Image in MATLAB

Figure 5.6: Output for Red Mango Image in MATLAB

Chapter 6

RESULTS AND DISCUSSION

In this system mangoes are graded in three types like Green Mango, Yellow Mango and Red Mango which are based on machine learning method. This system considers RGB values, Size and Shape of Mangoes. Posterior analysis is used to obtain good probability. This helps to train system to detect appropriate maturity of mangoes. This experiment is conducted based on Naive Bayes and Posterior Analysis Technique, to compare the performance based on accuracy and defective pixels. From the previous system, this system gives the more accuracy as Posterior Analysis is used. Here we are going to use MATLAB tool, of version MATLABR2018a.

A. Experimental Setup

The system is implemented using MATLAB tool on Windows platform. The Naive Bayes code is executed in the editor window of the MATLAB tool, and the program is to be run and if the errors are occurred, we need to debug the errors and again run the code. A GUI-formatted output window is displayed, here firstly training datasets are loaded and then the testing datasets of each image is compared with the training datasets and the output is to be observed such as RGB values, Diameter, Maturity prediction and class of the mangoes.

B. Datasets of Mangoes

Standard datasets of mango images are available on online, we have downloaded the datasets for different types of mango images from GitHub, Kaggle and Mendeleev Datasets. Some for-training dataset and some for testing. Table 6.1 shows details of mangoes database:

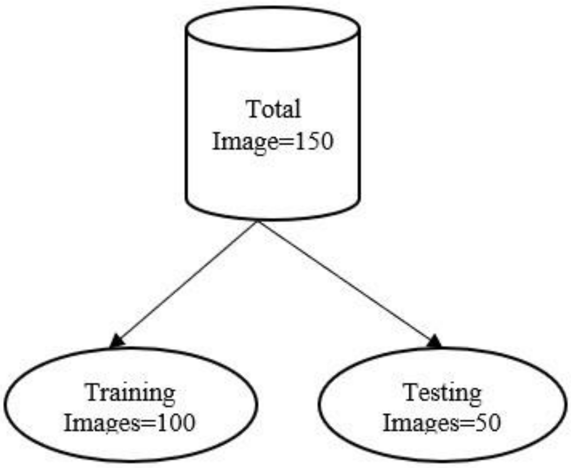

Figure 6.1: Datasets Classification

Table 6.1: Mangoes Database

Mangoes Type	Number of Training Images	Number of Testing Images	Total Images
Green Mango	100	50	150
Red Mango	100	50	150
Yellow Mango	100	50	150

C. Experimental Result

This system summarized here that the Naive Byes with Posterior Analysis Techniques can be applied on the images of mangoes for grade according to its maturity, size and Shape. In which image processing techniques has been used for exacting the features values of mangoes which is Shape, Size, Color and RGB values of mangoes in order to prepare a input to the Naive Byes with Posterior Analysis Techniques.

Table 6.2: Result of Classification of Mangoes

Sample Images	RGB Values	Diameter	Mango Class
	R=193.401 G=209.972 B=164.199	420.000	Green Mango
	R=230.585 G=169.550 B=167.141	340.000	Red Mango
	R=251.104 G=236.557 B=201.891	236.000	Yellow Mango

Chapter 7

CONCLUSION

The Mango Grading System determines Mangoes Maturity and Mangoes Quality in the form of Size, Shape and Surface Defects of Mangoes Images. By using various techniques increases the profit of vendors and time consumption is less.

This system is able to categorize the fruit depending on maturity and we consider the different features of mango fruit. For this we used machine learning methodology. Image pre-processing techniques are used to calculate the size of the Mango, surface of the mango is detected and shape is measured. The classification is done by adopting Naive Bayes and Posterior Analysis techniques. Here we can say that system gives better result as we used posterior analysis to improve the accuracy of training data set, also accurately measure the test image attribute. These techniques will train the images of Mangoes to predict the quality, maturity and color of the mango and the output obtained is graded Mango.

Our system works only for particular surface area consider detect the defective pixel. If other than that surface area defect is exists, we are not able detect it. For this in future need to consider the rotational view or the different surface view.

BIBLIOGRAPHY

[1] Dnyaneshwari Pise, J.D. Upadhye **"Grading of Harvested Mangoes Quality and Maturity Based on Machine Learning Techniques"**. Published in the year January – 2018.

[2] Dah-Jye Lee, James K. Archibald, Guangming Xiong **"Rapid Color Grading For Fruit Quality Evaluation Using Direct Mapping"** IEEE Transactions on Automation and Engineering Volume:08 in the year April-2011.

[3] Chandra Sekhar Nandi, Bipan Tudu, Chiranjib Koley **"Machine Vision Based Automatic Fruit Grading System Using Fuzzy Algorithm"** International Conference on Control, Instrumentation, Energy and Communication (CIEC) in the year 2014.

[4] Sapan Naik, Bankim Patel, Rashmi Pandey **"Shape, Size and Maturity Features Extraction with Fuzzy Classifier for non-destructive mango"** International Conference on Technological Innovations in ICT for Agriculture and Rural Development in the year 2015.

[5] Yogitha S and Sakthivel P **"A Distributed Computer Machine Vision System for Automated Inspection and Grading of Fruits"**.

[6] Kedar Patil, Shriniwas Kadam, Suraj Kale, Yogesh Rachetti, Kiran Jagtap, Dr. K.H. Inamdar **"Machine Vision Based Autonomous Fruit Inspection and Sorting"**, International Research Journal of Engineering and Technology (IRJET) Volume:03 in the year July- 2016.

[7] Ms. Seema Banot, Dr. P.M. Mahajan **"A Fruit Detecting and Grading System Based on Image Processing"**. International Journal of Innovative Research in Electronics Engineering Volume:04, in the year January-2016.

[8] Chandra Sekhar Nandi, Bipan Tudu and Chiranjib Koley, **"A Machine Vision Technique for Grading of Harvested Mangoes Based on Maturity and Quality"**. IEEE Sensor Journal in the year 2016.

[9] Mr. Sumit S Telang, Prof. S.M. Shirsath **"Fruit Quality Management Using Image Processing"**. International Conference on Ideas, Impact and Innovation in Mechanical Engineering Volume:05, in the year 2017.

[10] Dr. S. Usha, Dr. M. Karthik, R. Jenifer **“Automated Sorting and Grading of Vegetables Using Image Processing”**. International Journal of Engineering Research and General Science Volume:05, in the year 2017.

[11] Shwetha S Deulkar, Dr. Sunita S Barve **“Feature Based Fruit Quality Grading System Using Support Vector Machine”**. IEEE International Conference on recent trends in Electronics and Communication Technology in the year May-2017.

[12] L. Agilandeeswari, M. Prabukumar, Shubham Goel **“Automatic Grading System for Mangoes using Multiclass SVM Classifier”**. International Journal of Pure and Applied Mathematics Volume:116, in the year 2017.

[13] Lu Wang, Xin Tian, Anyu Li and Hanxiao Li **“Machine Vision Applications in Agricultural Food Logistics”**. International conference on Business Intelligence and Financial Engineering, in the year 2013.

[14] https://medium.com/analytics-vidhya/image-segmentation-techniques-using-digital-image-processing- machine-learning-and-deep-learning-ccf9e4589e94

[15]https://www.academia.edu/33112352/Identification_and_Classification_of_Mango_Fruits_Using_Image_Processing

[16] https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[17] https://en.wikipedia.org/wiki/Methodology

[18] https://www.kaggle.com/moltean/fruits

[19] https://github.com/Horea94/Fruit-ImageDataset/tree/master/Training/Mango%20Red

[20] https://github.com/Horea94/Fruit-Images-Dataset/tree/master/Test/Mango

[21] http://www.cofilab.com/portfolio/mangoesdb/

[22] https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

[23] https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/

[24] https://www.sciencedirect.com/topics/engineering/naive-bayes-classifier

[25]https://www.phon.ucl.ac.uk/courses/spsci/matlab/lect2.html#:~:text=The%20MATLAB%20Editor%20Window%20is, program%20to%20the%20command%20window

APPENDICES

```
function varargout = NaiveBayesClassifier(varargin)
 gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...f
                   'gui_OpeningFcn', @NaiveBayesClassifier_OpeningFcn, ...
                   'gui_OutputFcn',  @NaiveBayesClassifier_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
    clc;
      % set window potition (get_size_screen/gsl_)
    gsl_ = get(0,'ScreenSize');

end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

 % --- Executes just before NaiveBayesClassifier is made visible.
function NaiveBayesClassifier_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for NaiveBayesClassifier
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes NaiveBayesClassifier wait for user response (see UIRESUME)
warning('off','MATLAB:HandleGraphics:ObsoletedProperty:JavaFrame');
jframe=get(handles.NaiveBayesClassifier,'javaframe');
jIcon=javax.swing.ImageIcon('citrus-icon.png');
jframe.setFigureIcon(jIcon);

% --- Outputs from this function are returned to the command line.
function varargout = NaiveBayesClassifier_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

 % --- Executes on button press in NaiveBayesClassifier.
function trainingdata_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject=guidata(gcbo);
GetImageTraining=get(NaiveBayesClassifierProject.ImageTraining,'Userdata');

% determine the training data path
path_data_train=strrep(cd,...
    'Mangoes_Quality','Mangoimage\Data Training');
```

```
 lots_of_data_train_jn=7;
 lots_of_data_train_jl=3;
 lots_of_data_train_jm=5;
 lots_of_feature=4;
 lots_of_class=3;

% initialization of matrix dataset
dataset=zeros(lots_of_data_train_jn,lots_of_feature);

for i=1:(lots_of_data_train_jn+lots_of_data_train_jl+lots_of_data_train_jm)
        if(i<=lots_of_data_train_jn)
                filename=strcat(path_data_train,'\','MangoGreen',...
        num2str(i),'.jpg');
        class{i}='Green';
    elseif(i<=(lots_of_data_train_jn+lots_of_data_train_jl))
        filename=strcat(path_data_train,'\','MangoYellow',...
            num2str(i-lots_of_data_train_jn),'.jpg');
        class{i}='Yellow';
    else
        filename=strcat(path_data_train,'\','MangoRed',...
            num2str(i-(lots_of_data_train_jn+lots_of_data_train_jl)),'.jpg');
        class{i}='Red';
    end

    I = imread (filename);

    % merisize image size


    if(size(I,3)==4)
        I(:,:,1)=[]; % convert to I = [MxNx3]
    end

    % count mean Red, Green, Blue
    mean_red=mean(mean(I(:,:,1)));
    mean_green=mean(mean(I(:,:,2)));
    mean_blue=mean(mean(I(:,:,3)));
        axes(handles.ImageTraining);
    imshow(I);
        set(NaiveBayesClassifierProject.ImageTraining,'Userdata',I);

    SizeImageTraining=size(I);

StringSizeImageTraining=sprintf(strcat(num2str(SizeImageTraining(1)),'x',num2str(SizeImageTraining(2)),'x',num2str(SizeImageTraining(3))));
    StringSizeImageTraining = strrep(StringSizeImageTraining,'x',' x ');

    set(handles.SizeImageTraining,'String',StringSizeImageTraining);
        set(handles.NameImageTraining,'String',sprintf(strcat('Image Training (Color) ->',num2str(i))));

    %% create gray-scale image
```

```
    I_gray=Function_ColorToGray(I);
      axes(handles.ImageGray);
    imshow(I_gray);
      I_biner=zeros(size(I_gray,1),size(I_gray,2));
    I_biner(find(I_gray<255))=1;

    axes(handles.ImageBinary);
    imshow(I_biner);

    %% create max filter image from binary image
           windowing_size=5;
      max_filter_I_biner=Function_MaxFilterBiner_(I_biner,windowing_size);

      axes(handles.ImageMaxFilter);
      imshow(max_filter_I_biner);


    %% count diameter with unit length horizontal each pixel
      % determine index which contains the value 1
       [idx_,idy_]=find(max_filter_I_biner==1);
       diameter=idy_(numel(idy_))-idy_(1)+1;

      % replace pixel value
      I_red=I(:,:,1);
      I_red(:,idy_(1):(idy_(1)+10))=105;
      I_red(:,(idy_(numel(idy_))-10):idy_(numel(idy_)))=105;
      I(:,:,1)=I_red;
      I_green=I(:,:,2);
      I_green(:,idy_(1):(idy_(1)+10))=75;
      I_green(:,(idy_(numel(idy_))-10):idy_(numel(idy_)))=75;
      I(:,:,2)=I_green;

      I_blue=I(:,:,3);
      I_blue(:,idy_(1):(idy_(1)+10))=245;
      I_blue(:,(idy_(numel(idy_))-10):idy_(numel(idy_)))=245;
      I(:,:,3)=I_blue;


      axes(handles.ImageTraining);
      imshow(I);

    % collect the feature value
    dataset(i,:)=[mean_red mean_green mean_blue diameter];

end

class';
dataset;
 set(handles.NameImageTraining,'String',sprintf(strcat('Image Training (Color) DONE
!')));

merge_data={dataset,class'};
```

```
 % lots of data multiply with lots of feature
data_multiply_feature=(lots_of_data_train_jn+lots_of_data_train_jl+lots_of_data_train_j
m)*lots_of_feature;

% collect data feature R, G, B and Diameter
for i=1:data_multiply_feature
  dat_init{i}=num2str(merge_data{1}(i),'%.3f');
end

for j=1:(lots_of_data_train_jn+lots_of_data_train_jl+lots_of_data_train_jm)
  dat_init{data_multiply_feature+j}=char(merge_data{2}(j));
end

dat=reshape(dat_init,[(lots_of_data_train_jn+lots_of_data_train_jl+lots_of_data_train_jm
) (lots_of_feature+1)]);

set(NaiveBayesClassifierProject.DataTraining,'Userdata',dat);

%% insert data into dataset
t=uitable('Data', dat, 'ColumnName',...
      {'R (Red)', 'G (Green)', 'B (Blue)', 'D (Diameter)','Class (Mango)'},...
      'Position', [20 20 430 150]);



   set(NaiveBayesClassifierProject.dataset_all_feature_class,'Userdata',dataset);

% --- Executes during object creation, after setting all properties.
function SizeImageTraining_CreateFcn(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function SizeImageTesting_CreateFcn(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function NameImageTraining_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NameImageTraining (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

 % --- Executes during object creation, after setting all properties.
function uitabledataset_CreateFcn(hObject, eventdata, handles)
% --- Executes when entered data in editable cell(s) in uitabledataset.
function uitabledataset_CellEditCallback(hObject, eventdata, handles)
% --- Executes when selected cell(s) is changed in uitabledataset.
function uitabledataset_CellSelectionCallback(hObject, eventdata, handles)
% --- Executes on button press in testingdata.
function testingdata_Callback(hObject, eventdata, handles)
% data training of Citrus nipis (jn)
lots_of_data_train_jn=7;
 % data training of Citrus lemon (jl)
lots_of_data_train_jl=3;
 % data training of Citrus orange (jm)
lots_of_data_train_jm=5;
 lots_of_feature=4;
```

```
lots_of_class=3;

[basefilename,path]= uigetfile({'*.*'},'Open All Image File');
filename= fullfile(path, basefilename);

if sum(strfind(lower(basefilename), '.'))==0
else

   I_testing = imread (filename);

   % if I = [MxNx4]
   if(size(I_testing,3)==4)
      I_testing(:,:,1)=[]; % convert to I = [MxNx3]
   end

   %set(NaiveBayesClassifierProject.NaiveBayesClassifier,'CurrentAxes',
   axes(handles.ImageTesting);
   imshow(I_testing);

   % count mean Red, Green, Blue
   mean_red_testing=mean(mean(I_testing(:,:,1)));
   mean_green_testing=mean(mean(I_testing(:,:,2)));
   mean_blue_testing=mean(mean(I_testing(:,:,3)));

   set(NaiveBayesClassifierProject.var_mean_red_testing,...
      'String',num2str(mean_red_testing,'%.3f'));

   set(NaiveBayesClassifierProject.var_mean_green_testing,...
      'String',num2str(mean_green_testing,'%.3f'));

   set(NaiveBayesClassifierProject.var_mean_blue_testing,...
      'String',num2str(mean_blue_testing,'%.3f'));

   set(NaiveBayesClassifierProject.ImageTesting,'Userdata',I_testing);

   SizeImageTesting=size(I_testing);

StringSizeImageTesting=sprintf(strcat(num2str(SizeImageTesting(1)),'x',num2str(SizeImageTesting(2)),'x',num2str(SizeImageTesting(3))));
   StringSizeImageTesting = strrep(StringSizeImageTesting,'x',' x ');

   set(handles.SizeImageTesting,'String',StringSizeImageTesting);

   %% create gray-scale image
   I_gray_testing=Function_ColorToGray(I_testing);

   axes(handles.ImageGrayTesting);
   imshow(I_gray_testing);
   I_biner_testing=zeros(size(I_gray_testing,1),size(I_gray_testing,2));
   I_biner_testing(find(I_gray_testing<255))=1;

   axes(handles.ImageBinaryTesting);
```

```
imshow(I_biner_testing);

%% create max filter image from biner_testing image
  windowing_size=5;

max_filter_I_biner_testing=Function_MaxFilterBiner_(I_biner_testing,windowing_size);

  axes(handles.ImageMaxFilterTesting);
  imshow(max_filter_I_biner_testing);
   [idx_,idy_]=find(max_filter_I_biner_testing==1);
   diameter_testing=idy_(numel(idy_))-idy_(1)+1;

   set(NaiveBayesClassifierProject.var_diameter_testing,...
  'String',num2str(diameter_testing,'%.2f'));

% collect feature value
dataset(1,:)=[mean_red_testing mean_green_testing mean_blue_testing
diameter_testing];

if(isempty(strfind(basefilename, 'Yellow'))==0)
   class_testing{1}='Yellow';
elseif(isempty(strfind(basefilename, 'Red'))==0)
   class_testing{1}='Red';
elseif(isempty(strfind(basefilename, 'Green'))==0)
   class_testing{1}='Green';
else
   class_testing{1}='UnKnown';
end

class_testing{1}

set(NaiveBayesClassifierProject.classification_result,...
   'String',char(class_testing{1}));
% replace pixel value
   I_red=I_testing(:,:,1);
   I_red(:,idy_(1):(idy_(1)+10))=105;
   I_red(:,(idy_(numel(idy_))-10):idy_(numel(idy_)))=105;
   I_testing(:,:,1)=I_red;

   I_green=I_testing(:,:,2);
   I_green(:,idy_(1):(idy_(1)+10))=75;
   I_green(:,(idy_(numel(idy_))-10):idy_(numel(idy_)))=75;
   I_testing(:,:,2)=I_green;
   I_blue=I_testing(:,:,3);
   I_blue(:,idy_(1):(idy_(1)+10))=245;
   I_blue(:,(idy_(numel(idy_))-10):idy_(numel(idy_)))=245;
   I_testing(:,:,3)=I_blue;
   axes(handles.ImageTesting);
   imshow(I_testing);
   dataset=get(NaiveBayesClassifierProject.dataset_all_feature_class,'Userdata');

   Xrgbd=[mean_red_testing;mean_green_testing;mean_blue_testing;
```

```
    diameter_testing];

    Getmean_varian=zeros(2*lots_of_class,lots_of_feature);

    for i=1:lots_of_feature
      % set feature 1 for R, 2 for G, 3 for B, 4 for D
      feature_rgbd=dataset(:,i);

      % count mean_class_citrus_nipis(jn),_lemon(jl),_orange(jm)
      mean_feature_rgbd_jn=mean(feature_rgbd(1:lots_of_data_train_jn));
      mean_feature_rgbd_jl=mean(feature_rgbd(...
        (lots_of_data_train_jn+1):(lots_of_data_train_jn+lots_of_data_train_jl)));
      mean_feature_rgbd_jm=mean(feature_rgbd(...
        (lots_of_data_train_jn+lots_of_data_train_jl+1):...
        (lots_of_data_train_jn+lots_of_data_train_jl+lots_of_data_train_jm)));

      % count varian_class_citrus_nipis(jn),_lemon(jl),_orange(jm
      varian_feature_rgbd_jn=var(feature_rgbd(1:lots_of_data_train_jn));
      varian_feature_rgbd_jl=var(feature_rgbd(...
        (lots_of_data_train_jn+1):(lots_of_data_train_jn+lots_of_data_train_jl)));
      varian_feature_rgbd_jm=var(feature_rgbd(...
        (lots_of_data_train_jn+lots_of_data_train_jl+1):...
        (lots_of_data_train_jn+lots_of_data_train_jl+lots_of_data_train_jm)));

      Getmean_varian(:,i)=[mean_feature_rgbd_jn,mean_feature_rgbd_jl,...
        mean_feature_rgbd_jm,varian_feature_rgbd_jn,varian_feature_rgbd_jl,...
        varian_feature_rgbd_jm];
    end


  % count Probability of Prior
  P_Prior_jn=lots_of_data_train_jn/(lots_of_data_train_jn+...
  lots_of_data_train_jl+lots_of_data_train_jm);
  P_Prior_jl=lots_of_data_train_jl/(lots_of_data_train_jn+...
    lots_of_data_train_jl+lots_of_data_train_jm);
  P_Prior_jm=lots_of_data_train_jm/(lots_of_data_train_jn+...
    lots_of_data_train_jl+lots_of_data_train_jm);

  % initialization Probability value of Posterior
  P_Posterior_jn=1*P_Prior_jn
  P_Posterior_jl=1*P_Prior_jl
  P_Posterior_jm=1*P_Prior_jm

  % count Probability of Likelihood from RBGD feature
  for i=1:lots_of_feature
    mean_varian_RGBD=Getmean_varian(:,i);
    P_Likelihood_x_RGB_to_jn=...
      (1/sqrt(2*(22/7)*mean_varian_RGBD(4)))...
      *exp(-1*(((Xrgbd(i)-mean_varian_RGBD(1))^2)/(2*mean_varian_RGBD(4))))
    P_Posterior_jn=P_Posterior_jn*P_Likelihood_x_RGB_to_jn

    mean_varian_RGBD(5)
```

```
            mean_varian_RGBD(2)
            Xrgbd(i)
            P_Likelihood_x_RGB_to_jl=...
               (1/sqrt(2*(22/7)*mean_varian_RGBD(5)))...
               *exp(-1*(((Xrgbd(i)-mean_varian_RGBD(2))^2)/(2*mean_varian_RGBD(5))))
            P_Posterior_jl=P_Posterior_jl*P_Likelihood_x_RGB_to_jl

            P_Likelihood_x_RGB_to_jm=...
               (1/sqrt(2*(22/7)*mean_varian_RGBD(6)))...
               *exp(-1*(((Xrgbd(i)-mean_varian_RGBD(3))^2)/(2*mean_varian_RGBD(6))))
            P_Posterior_jm=P_Posterior_jm*P_Likelihood_x_RGB_to_jm
        end

        set(NaiveBayesClassifierProject.posterior_class_jn,...
        'String',strcat(num2str(P_Posterior_jn,'%.20f'),{'  ('},num2str(P_Posterior_jn),')'));
        set(NaiveBayesClassifierProject.posterior_class_jl,...
        'String',strcat(num2str(P_Posterior_jl,'%.20f'),{'  ('},num2str(P_Posterior_jl),')'));
        set(NaiveBayesClassifierProject.posterior_class_jm,...
        'String',strcat(num2str(P_Posterior_jm,'%.20f'),{'  ('},num2str(P_Posterior_jm),')'));

        All_P_Posterior=[P_Posterior_jn;P_Posterior_jl;P_Posterior_jm];

        [vmax_Posterior,idxmax_Posterior]=max(All_P_Posterior);

        Decision_Of_Classification='';
        if(idxmax_Posterior==1)
            Decision_Of_Classification='Green';
        elseif(idxmax_Posterior==2)
            Decision_Of_Classification='Yellow';
        else
            Decision_Of_Classification='Red';
        end

        set(NaiveBayesClassifierProject.classification_result,...
        'String',Decision_Of_Classification);

        set(NaiveBayesClassifierProject.var_class_testing,...
        'String',Decision_Of_Classification);

end
function var_mean_red_testing_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
var_mean_red_testing = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.var_mean_red_testing = var_mean_red_testing;

% --- Executes during object creation, after setting all properties.
function var_mean_red_testing_CreateFcn(hObject, eventdata, handles)
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function var_mean_green_testing_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
var_mean_green_testing = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.var_mean_green_testing = var_mean_green_testing;


% --- Executes during object creation, after setting all properties.
function var_mean_green_testing_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function var_mean_blue_testing_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
var_mean_blur_testing = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.var_mean_blue_testing = var_mean_blue_testing;

 % --- Executes during object creation, after setting all properties.
function var_mean_blue_testing_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function var_diameter_testing_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
var_diameter_testing = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.var_diameter_testing = var_diameter_testing;
% --- Executes during object creation, after setting all properties.
function var_diameter_testing_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function var_class_testing_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
var_class_testing = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function var_class_testing_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function posterior_class_jn_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
var_class_jn = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.var_class_jn = var_class_jn;
```

```
 % --- Executes during object creation, after setting all properties.
function posterior_class_jn_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function posterior_class_jl_Callback(hObject, eventdata, handles)
% hObject    handle to posterior_class_jl (see GCBO)
NaiveBayesClassifierProject = guidata(gcbo);
var_class_jl = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.var_class_jl = var_class_jl;

% --- Executes during object creation, after setting all properties.
function posterior_class_jl_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function posterior_class_jm_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
var_class_jm = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.var_class_jm = var_class_jm;

 % --- Executes during object creation, after setting all properties.
function posterior_class_jm_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function mean_class_jn_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
mean_class_jn = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.mean_class_jn = mean_class_jn;

% --- Executes during object creation, after setting all properties.
function mean_class_jn_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function varian_class_jn_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
varian_class_jn = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.varian_class_jn = varian_class_jn;

% --- Executes during object creation, after setting all properties.
function varian_class_jn_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function mean_class_jl_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
mean_class_jl = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.mean_class_jl = mean_class_jl;

% --- Executes during object creation, after setting all properties.
function mean_class_jl_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function varian_class_jl_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
varian_class_jl = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.varian_class_jl = varian_class_jl;

% --- Executes during object creation, after setting all properties.
function varian_class_jl_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function mean_class_jm_Callback(hObject, eventdata, handles)
NaiveBayesClassifierProject = guidata(gcbo);
mean_class_jm = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.mean_class_jm = mean_class_jm;

% --- Executes during object creation, after setting all properties.
function mean_class_jm_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function varian_class_jm_Callback(hObject, eventdata, handles)
%       str2double(get(hObject,'String')) returns contents of varian_class_jm as a double
NaiveBayesClassifierProject = guidata(gcbo);
varian_class_jm = str2double(get(hObject, 'String'));
NaiveBayesClassifierProject.varian_class_jm = varian_class_jm;

% --- Executes during object creation, after setting all properties.
function varian_class_jm_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function classification_result_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function classification_result_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all properties.
function ImageTraining_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ImageTraining (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
 % Hint: place code in OpeningFcn to populate ImageTraining
```