CHAPTER-1

AIM:

To find out the maturity stage of lemon based on the color with help of digital image processing that can be help the former to right time to harvesting and get on time profit and customer use.

INTRODUCTION

In most developing nations such as India, agriculture forms the major part of the country's economy. It is due to this fact that a lot of money is spent each year by governments across the world, for utilizing new technologies, brought to light new methodologies of farming. It is a well-known fact that the production of a good quality afford not only benefits consumers, who pre-eminent form the top end of the food chain, but also economically benefits the farmer who work hard to produce good yield. In a farmer's aspects, the better the quality, the more the income, since agricultural produce is not only used for direct consumption by people but also by the food industry, which use the same for the manufacture of several products. Lemons are spread mainly in the Mediterranean basin and represent the largest fruit source for human consumption. Postharvest losses, mainly due to diseases and metabolic disorders of fruits, can cause severe wastage, reaching 30 to 50% of the total production. ilind (2008), reported that lemons are used either as cut fruit or in juice form, based on the need and convenience. They are known to possess nutritive as well as medicinal values, mainly as rich source of vitamin C. Further, they contain other vitamins also such as vitamin B, riboflavin and minerals like calcium, phosphorous, magnesium besides proteins and carbohydrates. Lemons are known to reduce risk of heart diseases, cancer and also work as antiseptic, astringent, digestive stimulant etc. Preserving quality and extending shelf life are essential objectives for postharvest technological innovation, determined by the proper handling, treatment, storage and transport of harvested produce. Lemon fruits are particularly appreciated for their positive health benefits, representing an important source of bioactive compounds with high antioxidant activity, such as vitamin C, hydroxycinnamic acids and flavonoids. During season, fruits are collected from trees in bundles, and then sorted according to maturity level for transportation to different location. The harvest maturity stage in flounce the postharvest quality of different fruits during storage. Therefore, the effects of four different maturity stages (green, green-yellow and yellow stages) and cold storage periods (30, 60 and 90 days) at 10 °C.

Lemon maturity is divided into four cases based on the percentage on the ripeness.

CASE1: The green color of the lemon is indicated as the un-mature fruit.

CASE 2: Green–yellow color of the lemon will indicate the matured fruit.

CASE 3: Yellow color of the lemon will be indicated as over matured fruits.

In the present era, both fruit and vegetable markets are subject to choice. Ripeness checking and grading is a well-known process that takes place in all industries. The inspection done by humans might go wrong because of laziness, or inaccuracy. Due to this, the farmer might suffer by not getting the right amount of price for their goods. Presently the grading and ripeness checking devices are too costly and farmers are unable to afford them. In this study, specifically for the detection of ripe/unripe tomatoes with/without defects in the crop field, two distinct methods are described and compared from captured images by a camera mounted on a mobile robot. For ripeness checking, the RGB mean is calculated with a set of rules. However, For traditional methods, ripened tomatoes can be detected from the natural or solid background using color thresholding, while natural tomatoes can be detected using RGB color adjustment.

This experimental work aimed to develop an efficient approach for identifying the ripening stages of tomatoes. Using the acquired image, the RGB values of the tomato were processed by MATLAB and used to identify the stage of the tomato.

An approach for identifying the maturity of tomatoes using color and texture features of tomato images consists of the following phases.

- Image Acquisition
- Image Pre-processing
- Image Segmentation
- Feature Extraction
- Image Classification

1.1 INTRODUCTION TO IMAGE PROCESSING

Digital image processing is concerned with the processing of an image. Image processing is a method to perform operations on images like enhancing images, extracting text from images, detecting the edge of the image, and many other operations. In digital image processing, we take an image and convert that image into different forms. Like if we take a color image we can convert it into a grey image. In this, both the input and output are an image. Usually, the Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them.

Today, it is a rapidly growing technology. It forms a core research area within engineering and computer science disciplines too. Image processing has its wide applications in robotics, machine learning, neural networking, signal processing, the medical field, graphics and animations, and in many other fields.

1.2 DIGITAL IMAGE PROCESSING

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

1.3. INFORMATION ABOUT MATLAB

MATLAB is a high-performance language for technical computing. Using familiar mathematical notation, problems and solutions are represented in an environment that integrates computation, visualization, and programming.

Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The advantage is that you can solve many technical computing problems, especially those that involve matrix and vector formulations, much faster than in scalar, no interactive languages like C or FORTRAN. The name MATLAB stands for matrix laboratory. A core purpose of MATLAB was to make matrix software easily accessible to users. LINPACK and EISPACK are joint projects representing the state-of-the-art in matrices software. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science MATLAB is the tool of choice in the industry for high productivity research and development.

1.4 OBJECTIVES:

- Detection and quantification of lemons using color transformation.
- Checking the maturity of lemons and classifying un-mature mature and overmature LEMONs using color
- Helps the farmer to know the perfect stage of maturity.
- This project helps the farmer to identify the stage of lemon in horticulture.
- It also helps in marketability after maturing.
- It allows formers to know the perfect maturity stage for harvesting.
- It allows customers to buy a proper matured fruit according to their requirements.
- It reduces the time consumed for identifying the maturity stage.

CHAPTER-2

HARDWARE AND SOFTWARE REQUIREMENT

For the application to work efficiently, there need to be certain hardware components or other software resources to be present on computer. These prerequisites are known as system requirements. These requirements include the minimum processor speed, memory, and disk space required to install Windows. In almost all cases, you will want to make sure that your hardware exceeds these requirements to provide adequate performance for the services and applications running on the server.

Another form of requirements known as software requirements deals with defining software resources requirements and prerequisites that need to be installed on a computer to provide optimal functioning of the system.

2.1 HARDWARE SPECIFICATION

- Processor : Intel Core i3 or higher
- RAM : 2 GB or higher
- Hard Disk : 500 GB & above

2.2 SOFTWARE SPECIFICATION

- Windows 10 Operating System (32 bit)
- MATLAB, R2015a (8.5.0 1976 36) & above
- Weka Processor 3.8.6 version.
- Microsoft Excel 2007 & above

CHAPTER-3

LITERATURE SURVEY

1. Strano, M.C Reported that lemons are used either as cut fruit or in juice form, based on the need and convenience. They are known to possess nutritive as well as medicinal values, mainly as rich source of vitamin C. Further, they contain other vitamins also such as vitamin B, riboflavin and minerals like calcium, phosphorous, magnesium besides proteins and carbohydrates. Lemons are known to reduce risk of heart diseases, cancer and also work as antiseptic, astringent, digestive stimulant etc.

2. Gonzalez-Molina et al. (2010), mentioned that citrus genus is the most important fruit tree crop in the world and lemon is the third most important Citrus species. Several studies highlighted lemon as an important health-promoting fruit rich in phenolic compounds as well as vitamins, minerals, dietary fiber, essential oils and carotenoids. Lemon fruit has a strong coimmercial value -the value of the product in respect of the quality factors- for the fresh products market and food industry.

3. Khojastehnazhand et al. (2010), reported that the color and size are the most important features for accurate classification and/or sorting of citrus such as oranges, lemons and tangerines. The feasibility of using machine M Misr J. Ag. Eng., April 2015 - 773 - vision systems to improve product quality while freeing people from the traditional hand sorting of agricultural materials.

4. Narendra and Hareesh (2010), said that the lighting source type, location and color quality play an important role in bringing out a clear image of the object. Lighting arrangements are grouped into front- or back-lighting. Front lighting serves as illumination focusing on the object for better detection of external surface features of the product while backlighting is used for enhancing the background of the object. Light sources used include incandescent lamps, fluorescent lamps, lasers, X-ray tubes and infra-red lamps.

5. Omid et al. (2010), mentioned that the signals from fruit samples were captured by the cameras, transferred to the PC through the video capture cards, digitized and stored on the PC into three user-defined buffers in red, green, and blue color coordinates (RGB) for further analysis. In the HSI system, hue value is comparatively stable and the color of citrus can be determined by calculating the average Hue value for the fruit.

6. Elmasry et al. (2012), concluded that some external quality criteria, such as color, texture, size, and shape, are actually automated on industrial graders, but grading of fruits and vegetables according to the other appearance criteria, such as bruises, rottenness, and some other unobvious defects which present the same color and texture to the sound peel, or defects which are always confused with the stem-end and calyxes, is not yet efficient and consequently remains a manual operation.

3.1 EXISTING SYSTEM:

Existing systems have used complex algorithms which may consume more time to compute the results and in the existing system most of the algorithms calculate maturity based on the area of fruit and color but only a few researchers have done work on identifying maturity stage based on color of the fruit and in most of the studies they have used expensive digital cameras to capture the images and they have used the concepts of deep learning. This system would have cause heavy loss to the farmers in marketability and also in their economic life. In some systems by the use of chemicals the ripening stage was being detected

Limitations

- Difficult to understand the concept
- More expensive
- Less accuracy
- Time-consuming

3.2 PROPOSED SYSTEM:

In the proposed system we can use simple algorithms and steps to maturity identification of lemon are done on the basis of feature extraction and Segmentation. The identification of the maturity stage is done on the basis of color which is the first parameter evaluated by the consumers in this project to identify the maturity stage digital image processing technique is used as the identification is based on color and here we can use simple mobile phone cameras to capture images to give as input to the algorithm.

Advantages of the proposed system

- Easy to understand the concepts
- Less expensive
- More accuracy
- Less time consuming

CHAPTER-4

SOFTWARE REQUIREMENT SPECIFICATION

INTRODUCTION

This software requirements specification (SRS) document covers a brief introduction consisting of this document's purpose, scope, the definitions, acronyms, and abbreviations of the terms used in this document, and the overall organization of this document.

The next section gives the overall description of our system "Checking the maturity of lemon based on the color Values using DIP". This overall description includes a product perceptive, product function, user characteristics, constraints, assumption and dependencies, and apportioning of requirements. After that we provide the specific and modeling requirements, respectively.

PURPOSE

The purpose of this document is to present a detailed description of the Checking the maturity of lemon based on the color Values using DIP. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and clients.

SCOPE

Checking the maturity of lemon based on color Values can be implemented in agricultural sector to elude the huge loss of farmers and easily detect the maturity level of the lemon. Through this project the farmers can easily identify the age factor of the lemon and can get better yield and productivity in market and they can also be safe from the pre and post cultivation of the lemon crops

CHAPTER-5

SYSTEM DESIGN

FLOWCHART

A flowchart is a picture of the separate steps of a process in sequential order. Generic project plans can be used for describing various processes, such as manufacturing processes, administrative or service processes, or project plans. It's a common process analysis tool and one of the seven basic quality tools. The boxes represent process operations, and the arrows show how control flows between them. Data flows to calculate the possible values that a variable can hold at various points in a program, and determine how those values are propagated throughout the program. Flowcharts can help in identifying its essential steps and simultaneously offer the bigger picture of the process. This diagrammatic representation can give a step-by-step solution to give problem. Data flows are not typically represented in a flowchart, in contrast with data flow diagrams; rather, they are implied by the sequencing of operation. Flowcharts are used in analyzing, designing, documenting, or managing a process or program in various fields.

CONTROL FLOW DIAGRAM:

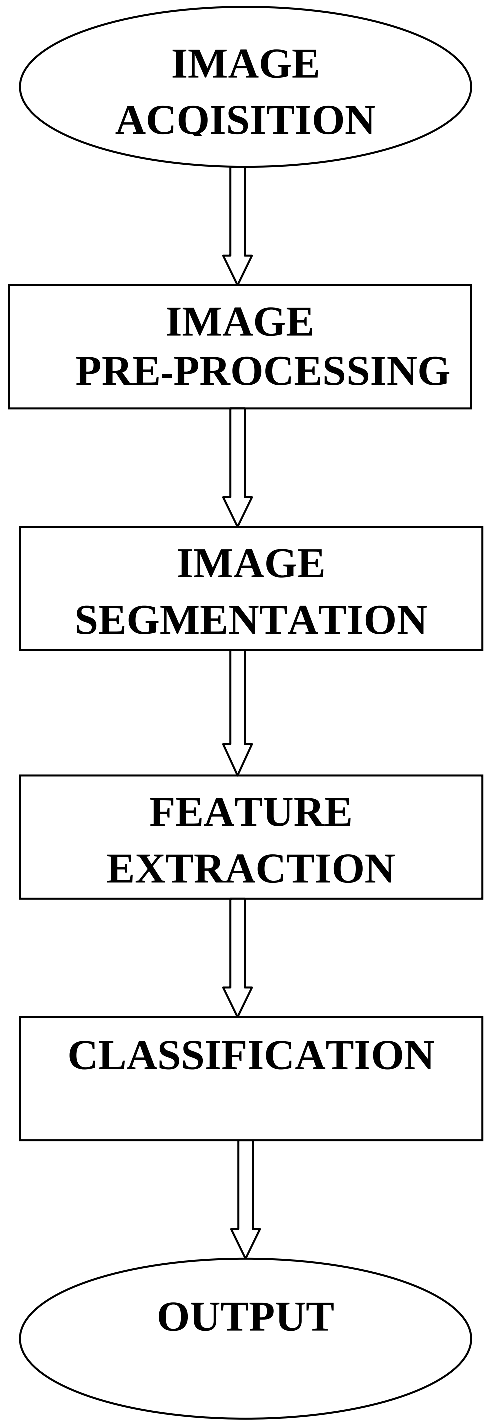

IMAGE ACQISITION

In image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. The first step in the workshop sequence is to get an image, as there can be no processing without a picture. The image that is acquired is completely unprocessed. The general goal of image acquisition is to convert an optical image (real-world data) into numerical data that can be manipulated later. By using an optical device, unprocessed images of objects or scenes are captured and converted into a processable format for processing and analysis.

IMAGE PRE-PROCESSING

Image pre-processing is an essential part of the process that can improve segmentation accuracy. The purpose of image preprocessing is to increase the contrast of an input image as well as enhance the visibility of a specified region. Image improvement, color-based transformation, noise reduction, and scaling are only a few of the procedures included. There are strengths and downsides to each of them that can be compared. Furthermore, it will emphasize segmentation until illness classification and detection are achieved.

IMAGE SEGMENTATION

Segmentation divides an image into its constituent regions or objects. The level of detail to which the subdivision carried depends on the problem using being solved. ie., segmentation should stop when the objects or regions of interest in an application have been detected.

In digital image processing and computer vision, image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. As a more precise definition, image segmentation involves assigning a label to each pixel in a given image so they can be grouped together based on certain features.

Image segmentation is based on some techniques are:

- Edge-based segmentation
- Threshold-based segmentation
- Region based segmentation

For image segmentation we can use the some masks like Robert, Sobel, Prewitt, and Canny and also we do detection processes. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes.

APPLICATIONS

Some of the practical applications of image segmentation are:

• Content-based image retrieval

• Machine vision

• Medical imaging, including volume rendered images from computed tomography and magnetic resonance imaging.

- Locate tumors and other pathologies
- Measure tissue volumes
- Diagnosis, study of anatomical structure
- Surgery planning
- Virtual surgery simulation o Intra-surgery navigation

• Object detection

- Pedestrian detection
- Face detection
- Brake light detection
- Locate objects in satellite images

• Recognition Tasks o Face recognition

- Fingerprint recognition
- Iris recognition

• Traffic control systems

• Video surveillance

• Video object co-segmentation and action localization

Several general-purpose algorithms and techniques have been developed for image segmentation. To be useful, these techniques must typically be combined with a domain's specific knowledge in order to effectively solve the domain's segmentation problems.

THRESHOLDING

Image thresholding is a simple form of image segmentation. It is a way to create a binary image from a gray scale or full-color image. This is typically done in order to separate "object" or foreground pixels from background pixels to aid in image processing.

The key of this method is to select the threshold value (or values when multiple levels are selected). Several popular methods are used in industry including the maximum entropy method, balanced histogram thresholding, Otsu's method (maximum variance), and k-means clustering.

Recently, methods have been developed for thresholding computed tomography (CT) images. The key idea is that, unlike Otsu's method, the thresholds are derived from the radiographs instead of the (reconstructed) image.

New methods suggested the usage of multi-dimensional fuzzy rule-based nonlinear thresholds. In these works decision over each pixel's membership to a segment is based on multidimensional rules derived from fuzzy logic and evolutionary algorithms based on image lighting environment and application.

CLUSTERING METHODS

The k-means algorithm is an iterative technique that is used to partition an image into k clusters. The basic algorithm is

1. Pick K cluster centers, either randomly or based on some heuristic method, for example K-means++

2. Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center

3. Re-compute the cluster centers by averaging all of the pixels in the cluster

4. Repeat steps 2 and 3 until convergence is attained (i.e. no pixels change clusters)

In this case, distance is the squared or absolute difference between a pixel and a cluster center. The difference is typically based on pixel color, intensity, texture, and location, or a weighted combination of these factors. K can be selected manually, randomly, or by a heuristic. This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K.

COMPRESSION-BASED METHODS

Compression based methods postulate that the optimal segmentation is the one that minimizes, over all possible segmentations, the coding length of the data. The connection between these two concepts is that segmentation tries to find patterns in an image and any regularity in the image can be used to compress it. The method describes each segment by its texture and boundary shape. Each of these components is modeled by a probability distribution function and its coding length is computed as follows:

1. The boundary encoding leverages the fact that regions in natural images tend to have a smooth contour. This prior is used by Huffman coding to encode the difference chain code of the contours in an image. Thus, the smoother a boundary is, the shorter coding length it attains.

2. Texture is encoded by loss compression in a way similar to minimum description length (MDL) principle, but here the length of the data given the model is approximated by the number of samples times the entropy of the model. The texture in each region is modeled by a multivariate normal distribution whose entropy has a closed form expression. An interesting property of this model is that the estimated entropy bounds the true entropy of the data from above. This is because among all distributions with a given mean and covariance, normal distribution has the largest entropy. Thus, the true coding length cannot be more than what the algorithm tries to minimize.

For any given segmentation of an image, this scheme yields the number of bits required to encode that image based on the given segmentation. Thus, among all possible segmentations of an image, the goal is to find the segmentation which produces the shortest coding length. This can be achieved by a simple agglomerative clustering method. The distortion in the lossy compression determines the coarseness of the segmentation and its optimal value may differ for each image. This parameter can be estimated heuristically from the contrast of textures in an image. For example, when the textures in an image are similar, such as in camouflage images, stronger sensitivity and thus lower quantization is required.

FEATURE EXTRACTION

Feature extraction is a part of the dimensionality reduction process. In which an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. A main and most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process them. So feature extraction basically helps to get the best feature from that big data set by select and/or combine variables into features, effectively reducing the amount of data these features are easy to process, but still able to describe the actual data set with the accuracy and originality.

Why is this Useful?

Process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction can also reduce the amount of the redundant data for a given analysis. In addition, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

Practical Uses of Feature Extraction

• Auto encoders

The purpose of auto encoders is unsupervised learning of efficient data coding. Feature extraction is used here to identify key features in the data for coding by learning from the coding of the original data set to derive new ones.

• Bag-of-words

A technique for natural language processing that extracts the words (features) used in a sentence, document, website, etc. and classifies them by frequency of use. This technique can also be applied to image processing

• Image processing

Algorithms are used to detect features such as shaped, edges, or motion in a digital image or video. Applications of feature extraction

• You can automate identifying and extracting features from large volumes of text data and create a summary of all unique features or a combination of features with a text extraction solution Brands and marketers need to understand the needs of their target audience. Use text extraction to gain valuable insights and create products with a compelling set of features.

• Recommendation systems can detect other products that contain similar entities by automatically identifying entities in a product specification.

• You can create different groups of features by clustering similar or related features.

• Identifying new product features or new segments of customers and develop more personalized and appealing products.

• By making it simple to find, organize, and access relevant stuff, you may learn new things.

CLASSIFICATION

Image classification is the process of categorizing and labeling groups of pixels or vectors within an image based on specific rules. Digital image classification uses the spectral information represented by the digital numbers in one or more spectral bands, and attempts to classify each individual pixel based on this spectral information. This type of classification is termed spectral pattern recognition.

Random tree classifier

One way to overcome this limitation is to produce many variants of a single decision tree by selecting every time a different subset of the same training set in the context of randomization-based ensemble methods. Random Forest Trees (RFT) is a machine learning algorithm based on decision trees. Random Trees (RT) belong to a class of machine learning algorithms which does ensemble classification. The term ensemble implies a method which makes predictions by averaging over the predictions of several independent base models.

The fundamental principle of ensemble methods based on randomization "is to introduce random perturbations into the learning procedure in order to produce several different models from a single learning set L and then to combine the predictions of those models to form the prediction of the ensemble. IN other words, "significant improvements in classification accuracy have resulted from growing an ensemble of trees and letting them vote for the most popular class. In order to grow these ensembles, often random vectors are generated that govern the growth of each tree in the ensemble".

There are three main choices to be made when constructing a random tree. These are

(1) The method for splitting the leafs and fruits,
(2) The type of predictor to use in each leaf and fruit, an
(3) The method for injecting randomness into the trees.

Some of the key features of Random Trees are:

- It is an excellent classifier--comparable in accuracy to support vector machines.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when up to 80% of the data are missing.
- It has a method for balancing error in unbalanced class population data sets.
- Generated forests can be saved for future use on other data.
- It gives estimates of what variables are important in the classification.
- Output is generated that gives information about the relation between the variables and the classification.

CHAPTER-6

SYSTEM IMPLEMENTATION

Implementation is a phase that has to be accomplished after, all the systems development process has been completed. An important aspect of a system analyst job is to make sure that new design is implemented according to the established standards. Once the system design phase is over, the next stage is to implement and to find out the efficiency and effectiveness of the system.

Step involved in the implementation process are:

- Verification of testing process.
- Implementing the tool.
- Working with actual data.
- User acceptance.
- User adaptability

VERIFICATION OF TESTING PROCESS

The team who has developed the system would have carried out the testing before it is actually implemented and it will be forwarded to other programmer who has not participated in the development of the project. This is done to make sure that the developed system is error free.

IMPLEMENTING THE TOOL

In order to implement this system, all the necessary hardware and software specifications must be available.

WORKING WITH ACTUAL DATA

After implementing this, it is subjected to a actual working circumstance. The user enters the actual input wherever necessary. This is the final phase where the life cycle of the project ends. The continuity goes on and on. This makes the system non-volatile. The users could conveniently work on this system forever until this needs a change for the system.

USER ACCEPTANCE

When there is a need for the user, to make a change in the system or if the user would like to know additional information related to this, then they would suggest for up gradation through feedback. This application has been completed so as to satisfy the user by meeting all their possible needs and has been confirmed that this system works as its best.

USER ADAPTABILITY

Even well designed and technically elegant system can fail because of the way there are operated and used. Therefore the quantity of training the personnel involved with system in various capacities helps the successful implementation system.

CHAPTER-7

SOURCE CODE

1. CODING USED IN THE PROJECT FOR INDIVIDUAL FRUIT

```
close all;
clear all;
clc;
% Acquiring the image from folder
im = imread('ut17.jpeg');
%Resizing the images
[~, ~, ~] = size(im);
p=imresize(im,[250,250]);
subplot(2,3,1),
imshow(p),
title('original image');

%Converting the original image to gray scale
I = rgb2gray(p);
subplot(2,3,2),
imshow(I),
title('Gray Image')

%applying binary gradient Mask
[~,threshold] = edge(I,'sobel');
fudgeFactor = 0.5;
BWs = edge(I,'sobel',threshold * fudgeFactor);
subplot(2,3,3),
imshow(BWs),
title('Binary Gradient Mask');
% To fill the holes
BWdfill = imfill(BWs,'holes');
subplot(2,3,4),
imshow(BWdfill),
title({'Binary Image with','Filled Holes'});
% to filter the image
[r,c]=size(BWdfill);
fim=medfilt2(BWdfill);
subplot(2,3,5),
imshow(fim),
```

```
title({'Filtered and',' Segmented Image'});

% converting the segmented image to color image
colimg=uint8(zeros(r,c));
for i=1:r
    for j=1:c
        for k=1:3
            if  fim(i,j)==0
                colimg(i,j,k)=uint8(fim(i,j));
            else
                colimg(i,j,k)=uint8(p(i,j,k));
            end
        end
    end
end
subplot(2,3,6),
imshow(colimg),
title('color image');
I1=colimg;
grim=rgb2gray(I1);
c1=0;
[r,c]=size(grim);
for i=1:r
    for j=1:c
        if(grim(i,j)>0)
            c1=c1+1;
        end
    end
end
imR=colimg;
imR(:,:,2:3)=0;
imG=I1;
imG(:,:,1:2:3)=0;
imB=I1;
imB(:,:,1:2)=0;
R(1)=((sum(imR(:)))/c1);
%R(2)=mean(imR(:));
% R(3)=var(double(imR(:)));
G(1)=((sum(imG(:)))/c1);
% G(2)=std2(imG(:));
```

```
% G(3)=var(double(imG(:)));
B(1)=((sum(imB(:)))/c1);
% B(2)=std2(imB(:));
% B(3)=var(double(imB(:)));

%Feature Extraction
GI=rgb2gray(I1);
glcm = graycomatrix(GI);
R1 = graycoprops(glcm);
R2(1) = R1.Contrast;
R2(2) = R1.Correlation;
R2(3) = R1.Energy;
R2(4) = R1.Homogeneity;
R2(5)=entropy(GI);
fprintf('Rmean\tGmean\tBmean\tContrast\tCorrelation\tEnergy\tHomogeneity\tTexture\n')
fprintf('%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\n',R(1),G(1),B(1),R2(1),R2(2),R2(3),R2(4),R2(5));
textureval = R1.Homogeneity*10;
dispTextureValue
disp(textureval/10);

fprintf('\n........SEGMENTATION THROUGH COLOUR VALUES........\n\n');
greenpoints=colimg(:,:,1)>=110&colimg(:,:,2)<=255& colimg(:,:,3)<=100;
percentgreen=100*(sum(sum(greenpoints))/(size(colimg,1)*size(colimg,2)));
fprintf('image has %d green pixels\n\n',sum(sum(greenpoints)));
fprintf('image is %.2f percent green\n\n',percentgreen);

%displaying output based on threshold value
if (sum(sum(percentgreen))>0&&sum(sum(percentgreen))<6)
fprintf(Lemon is unmature and not to be Harvest\n');
end
if  (sum(sum(percentgreen))>6&&sum(sum(percentgreen))<15)
fprintf(Lemon is mature and eligible to Harvest\n');
end
 if  (sum(sum(percentgreen))>15&&sum(sum(percentgreen))<30)
fprintf(Lemon is overmature and Harvest fast\n');
 end
```

2.CODING USED IN THE PROJECT FOR ALL TOMATO FRUIT

```
clc;
clear all;
y=inputdlg('Enter The Excel File name',...  'Sigma Input',[1 40]);
sigma_input=y{:};
disp(sigma_input);
extt='.xls';
filename1=strcat(sigma_input,extt);
disp(filename1);
close all;
fid3=fopen(filename1, 'w');
fprintf(fid3,'Image\tRMean\tGMean\tBMean\tContrast\tCorrelation\tEnergy\tHomogeneity\t
Entropy\tTexture\tPercentage\n');
Dir=uigetdir(pwd,'select input folder');
```

% Read images from Images folder

```
Imgs = dir(fullfile(Dir, '*.jpeg'));
for q=1:length(Imgs)
FileName=fullfile(Dir, Imgs(q).name);
im=imread(FileName);
%im = imread('G:\pics\1.jpeg');
[~, ~, ~] = size(im);
p=imresize(im,[250,250]);
%figure,imshow(p),title('original image');
```

%Converting the original image to gray scale

```
I = rgb2gray(p);
%%figure,imshow(I),title('Gray Image')
%applying binary gradient Mask
[~,threshold] = edge(I,'sobel');
fudgeFactor = 0.5;
BWs = edge(I,'sobel',threshold * fudgeFactor);
%%figure,imshow(BWs),title('Binary Gradient Mask');
% to fill the holes BWdfill = imfill(BWs,'holes');
%%figure,imshow(BWdfill)
%title('Binary Image with Filled Holes');
% to filter the image [r,c]=size(BWdfill);
fim=medfilt2(BWdfill);
%%figure,imshow(fim),title('Filtered & Segmented Image');
```

% converting the segmented image to color image

```
colimg=uint8(zeros(r,c));
for i=1:r
    for j=1:c
        for k=1:3
            if  fim(i,j)==0
                colimg(i,j,k)=uint8(fim(i,j));
            else
                colimg(i,j,k)=uint8(p(i,j,k));
            end
        end
    end
end
figure,
imshow(colimg),
title('color image');
```

%maturity identification

```
I1=colimg;
grim=rgb2gray(I1);
c1=0; [r,c]=size(grim);
for i=1:r
    for j=1:c
        if(grim(i,j)>0)
            c1=c1+1;
        end
    end
end
imR=colimg;
imR(:,:,2:3)=0;
imG=I1;
imG(:,:,1:2:3)=0;
imB=I1;
imB(:,:,1:2)=0;
R(1)=((sum(imR(:)))/c1);
%R(2)=mean(imR(:));
% R(3)=var(double(imR(:)));
G(1)=((sum(imG(:)))/c1);
% G(2)=std2(imG(:));
% G(3)=var(double(imG(:)));
```

```
B(1)=((sum(imB(:)))/c1);
% B(2)=std2(imB(:));
% B(3)=var(double(imB(:)));
%Feature Extraction GI=rgb2gray(I1);
glcm = graycomatrix(GI);
R1 = graycoprops(glcm);
R2(1) = R1.Contrast;
R2(2) = R1.Correlation;
R2(3) = R1.Energy;
R2(4) = R1.Homogeneity;
R2(5)=entropy(GI);
texture=R2(4)*100;
greenpoints=colimg(:,:,1)>=110&colimg(:,:,2)<=255&colimg(:,:,3)<=100;
percentgreen=100*(sum(sum(greenpoints))/(size(colimg,1)*size(colimg,2)));
fprintf(fid3,'%s\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\n',FileName,R(1),G(1),B(1),R2(1
),R2(2),R2(3),R2(4),R2(5),texture,percentgreen);
end
```

CHAPTER-8

SCREENSHORT

A = imread(filename) :Reads The Image From The File Specified By Filename, Inferring The Format Of The File From Its Contents. If Filename Is A Multi-Image File, Then Imread Reads The First Image In The File.

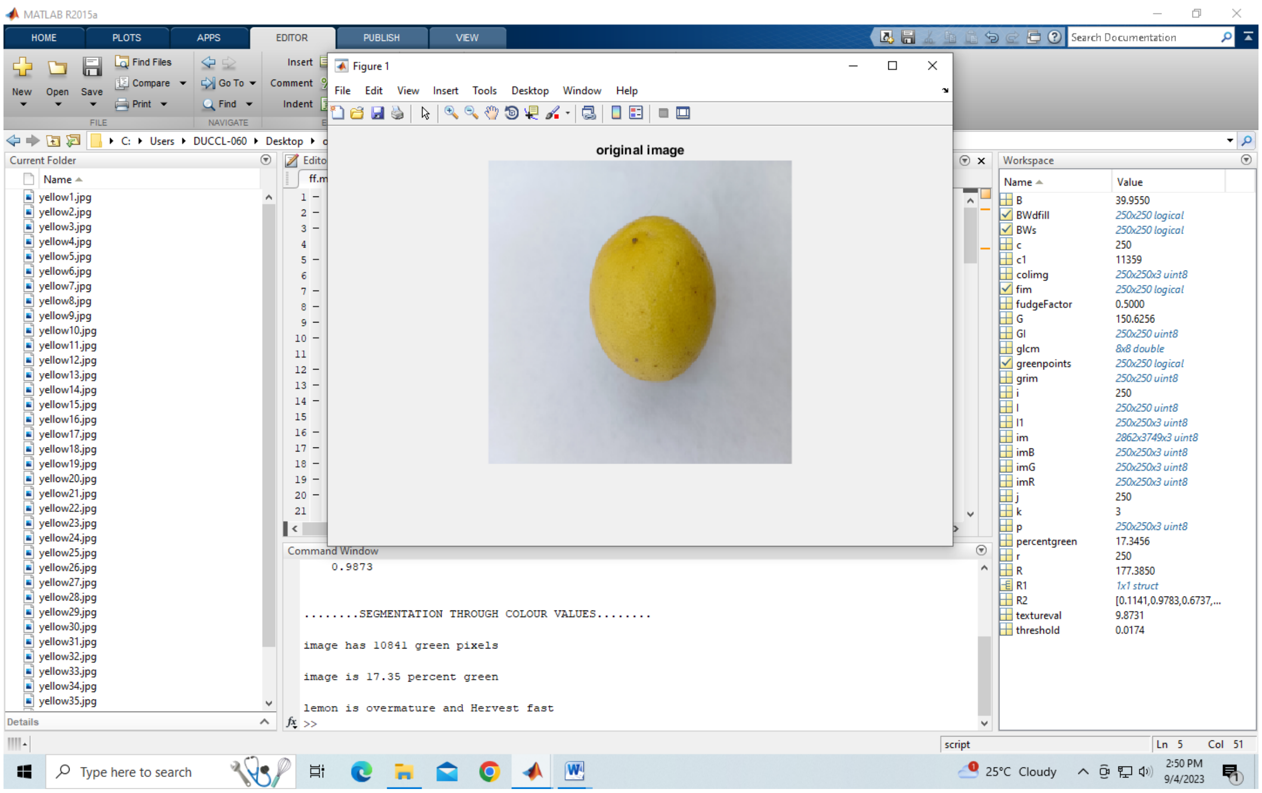

Figure1.1 original image

Gray Image: Grayscale is a range of monochromatic shades from black to white. Therefore, a grayscale image contains only shades of gray and no color.

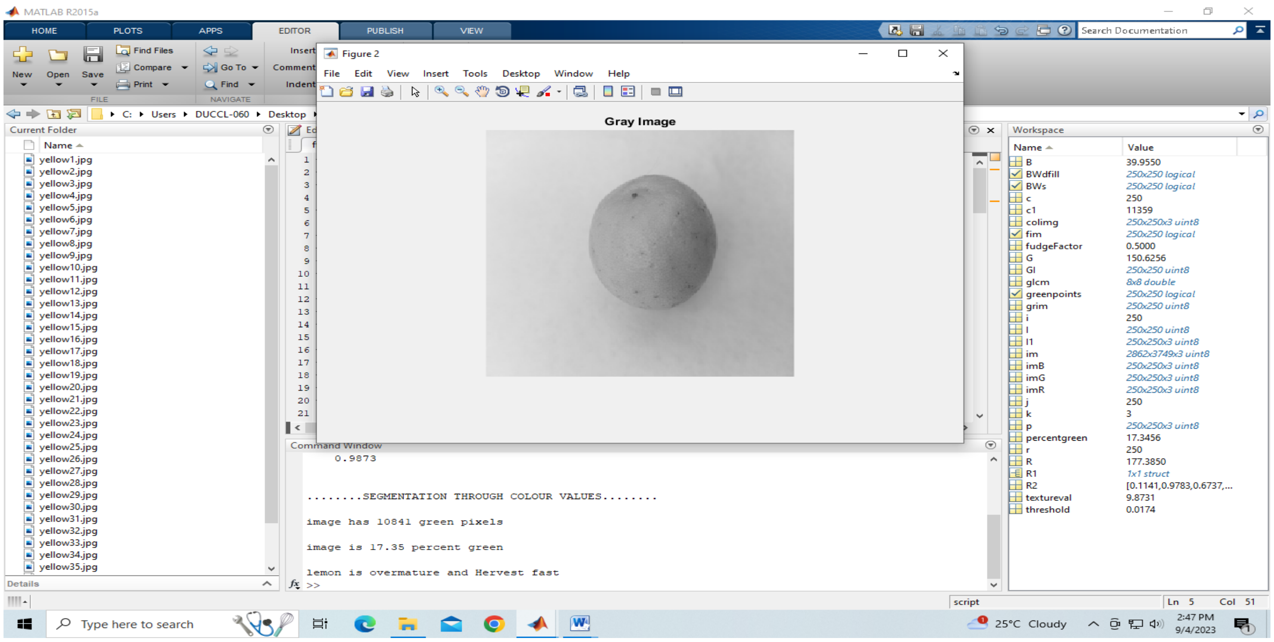

Figure 1.2 gray image.

Gradient Image:-An image gradient is a directional change in the intensity or color in an image. The gradient of the image is one of the fundamental building blocks in image processing.

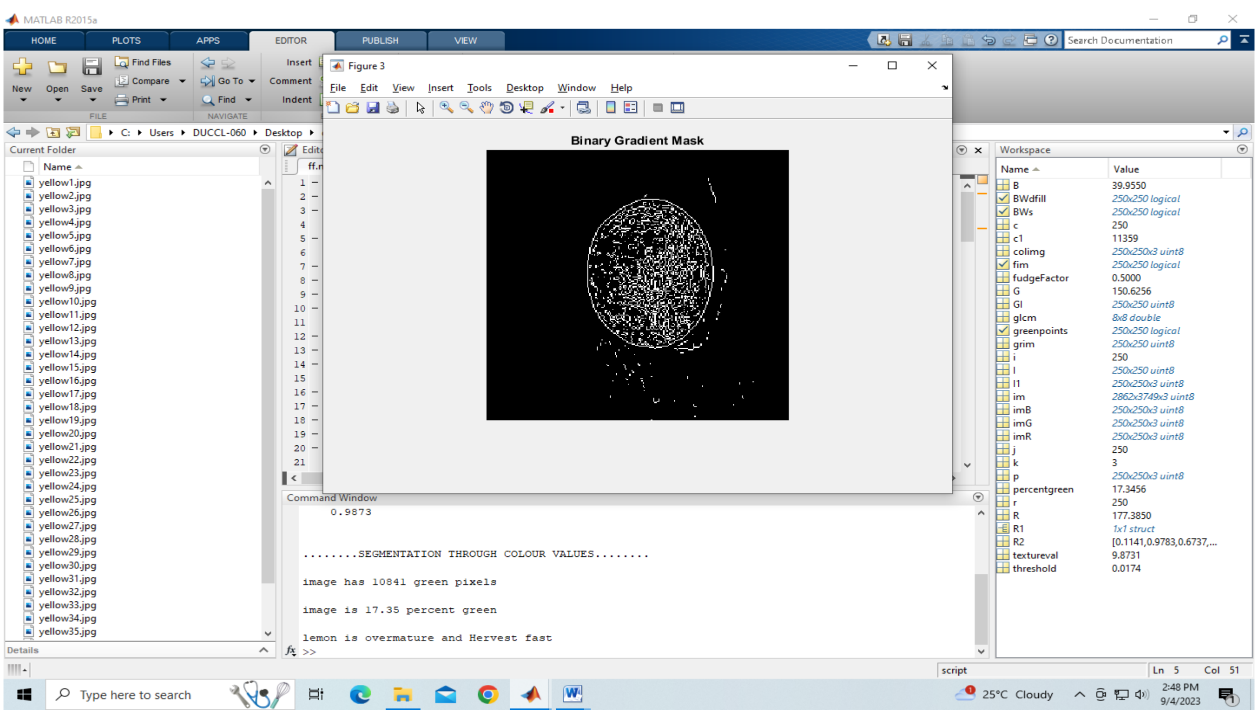

Figure 1.3 binary gradient image.

Filled holes: I2=imfill(I,conn) fills holes in the grayscale image I, where conn specifies the connectivity. BW2 =imfill(BW) displays the binary image BW on the screen and lets you define the region to fill by selecting points interactively with the mouse. To use this syntax, BW must be a 2-D image.

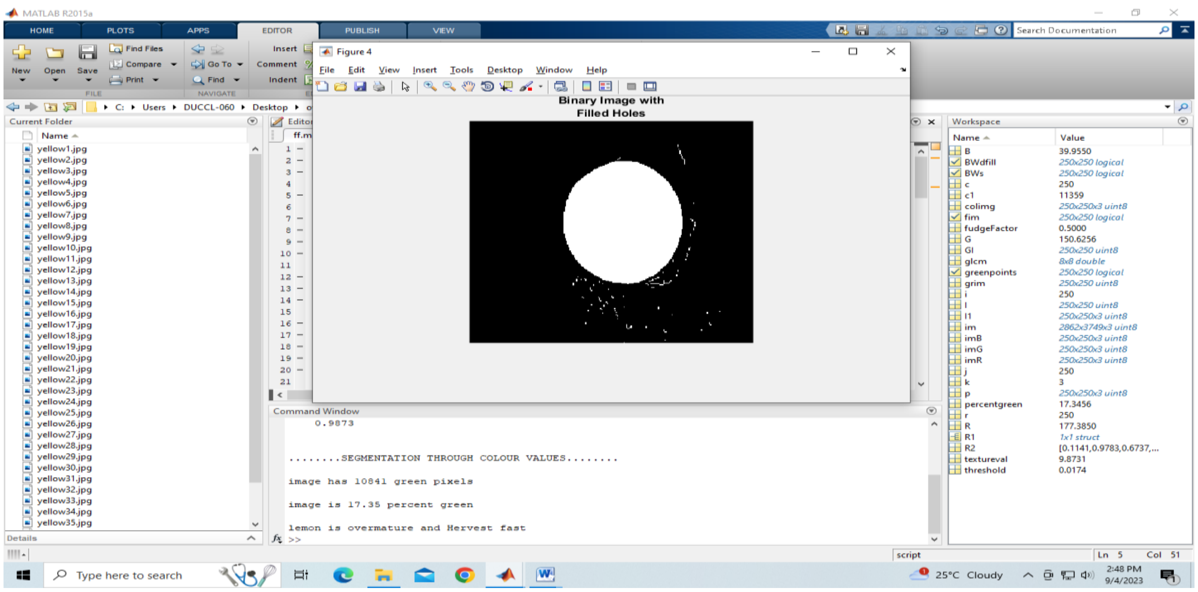

Figure1.4 binary image with filled holes

Filtered & Segmented Image: Image Segmentation is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, and thus analyzing the image becomes simpler.

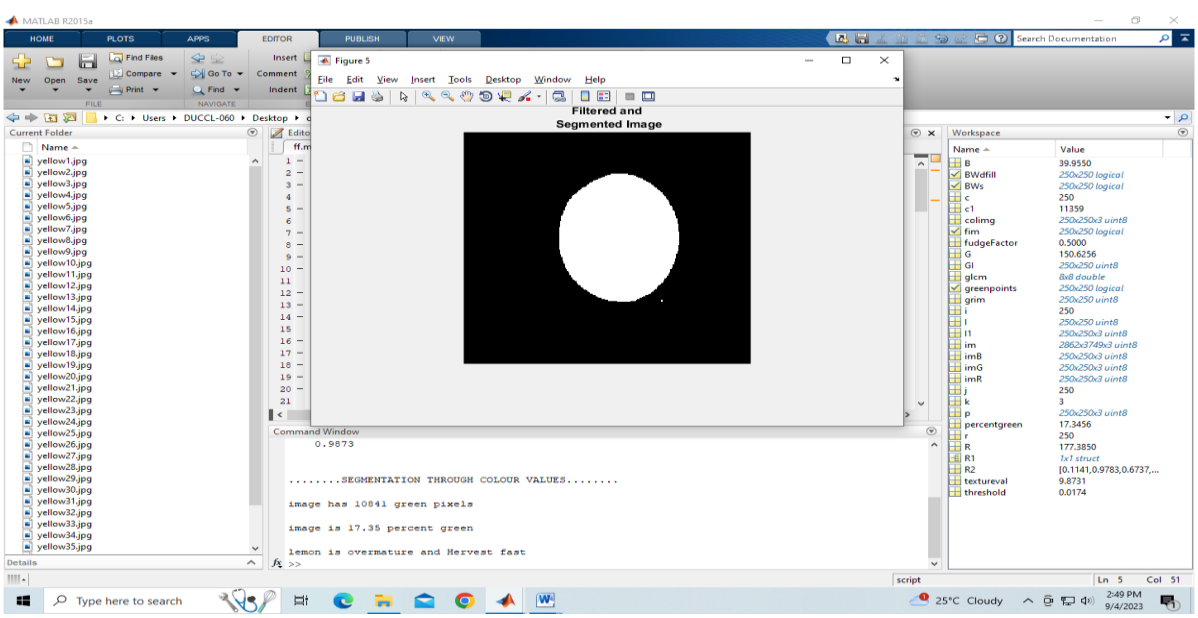

Figure1.5 filtered and segmented image

Color Image: Image segmentation is a process of assigning a label to every pixel in an image such that pixels with same label share certain visual characteristics. Sometimes it becomes necessary to calculate the total number of colors from the given RGB image to quantize the image.

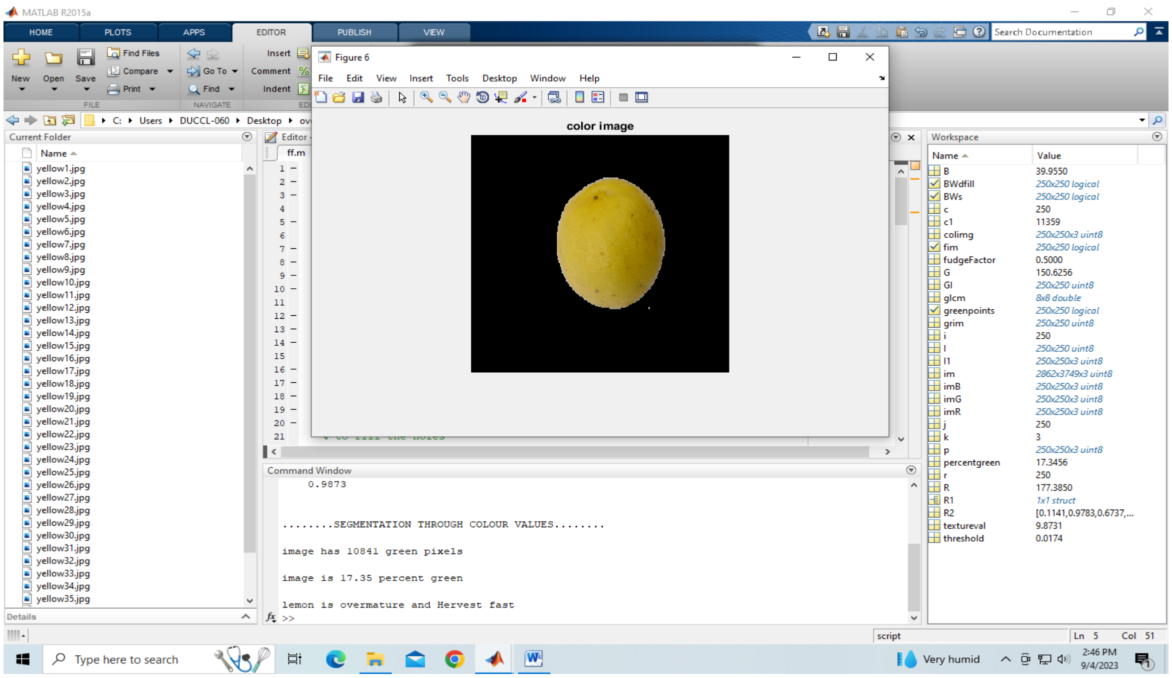

Figure1.6. Segmented output image

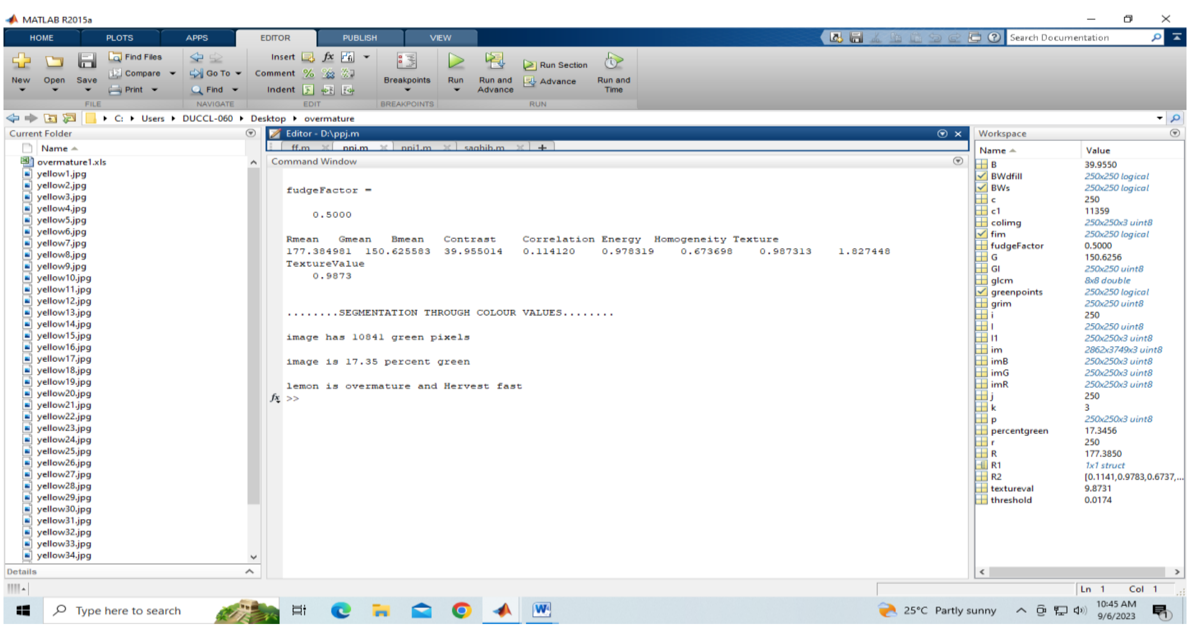

Figure 1.7.Text massage showing the over-mature of lemon

TESTING:

The image is identified and the color and texture values are displayed in text.

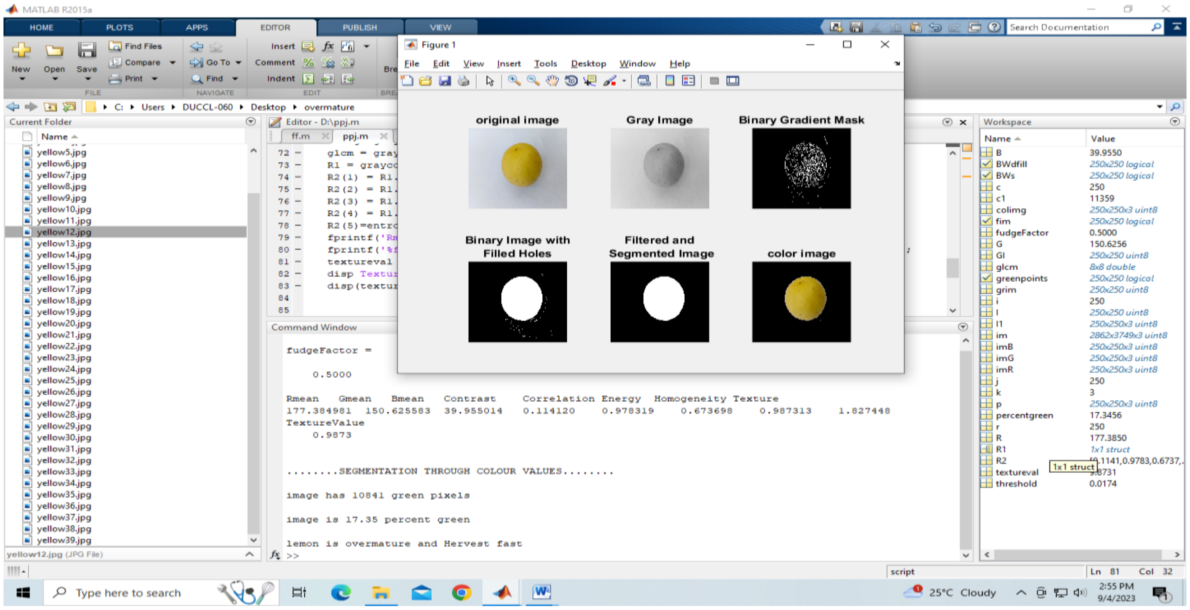

Figure 1.8 color image

EXCEL SHEET

The values of maturity of lemon are extracted using the features of the lemon i.e. based on the color. The values contain the RMean, GMean, BMean, Contrast, Correlation, Energy, Homogeneity and Entropy values of the segmented color image.

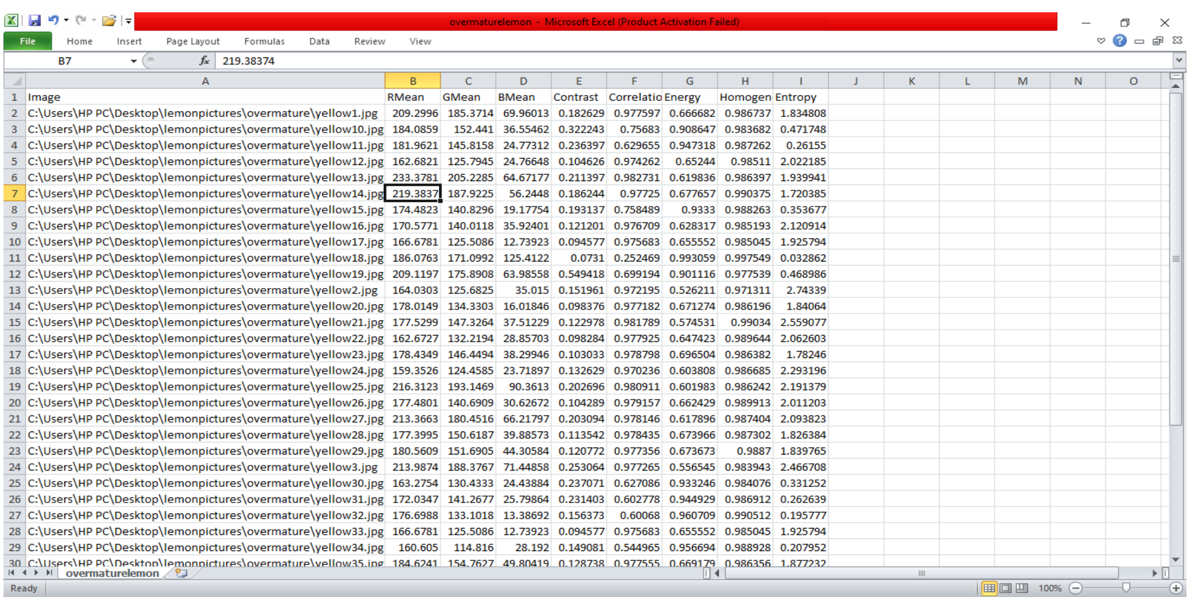

Image	RMean	GMean	BMean	Contrast	Correlatio	Energy	Homogen	Entropy
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow1.jpg	209.2996	185.3714	69.96013	0.182629	0.977597	0.666682	0.986737	1.834808
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow10.jpg	184.0859	152.441	36.55462	0.322243	0.75683	0.908647	0.983682	0.471748
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow11.jpg	181.9621	145.8158	24.77312	0.236397	0.629655	0.947318	0.987262	0.26155
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow12.jpg	162.6821	125.7945	24.76648	0.104626	0.974262	0.65244	0.98511	2.022185
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow13.jpg	233.3781	205.2285	64.67177	0.211397	0.982731	0.619836	0.986397	1.939941
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow14.jpg	219.3837	187.9225	56.2448	0.186244	0.97725	0.677657	0.990375	1.720385
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow15.jpg	174.4823	140.8296	19.17754	0.193137	0.758489	0.9333	0.988263	0.353677
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow16.jpg	170.5771	140.0118	35.92401	0.121201	0.976709	0.628317	0.985193	2.120914
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow17.jpg	166.6781	125.5086	12.73923	0.094577	0.975683	0.655552	0.985045	1.925794
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow18.jpg	186.0763	171.0992	125.4122	0.0731	0.252469	0.993059	0.997549	0.032862
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow19.jpg	209.1197	175.8908	63.98558	0.549418	0.699194	0.901116	0.977539	0.468986
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow2.jpg	164.0303	125.6825	35.015	0.151961	0.972195	0.526211	0.971311	2.74339
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow20.jpg	178.0149	134.3303	16.01846	0.098376	0.977182	0.671274	0.986196	1.84064
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow21.jpg	177.5299	147.3264	37.51229	0.122978	0.981789	0.574531	0.99034	2.559077
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow22.jpg	162.6727	132.2194	28.85703	0.098284	0.977925	0.647423	0.989644	2.062603
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow23.jpg	178.4349	146.4494	38.29946	0.103033	0.978798	0.696504	0.986382	1.78246
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow24.jpg	159.3526	124.4585	23.71897	0.132629	0.970236	0.603808	0.986685	2.293196
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow25.jpg	216.3123	193.1469	90.3613	0.202696	0.980911	0.601983	0.986242	2.191379
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow26.jpg	177.4801	140.6909	30.62672	0.104289	0.979157	0.662429	0.989913	2.011203
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow27.jpg	213.3663	180.4516	66.21797	0.203094	0.978146	0.617896	0.987404	2.093823
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow28.jpg	177.3995	150.6187	39.88573	0.113542	0.978435	0.673966	0.987302	1.826384
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow29.jpg	180.5609	151.6905	44.30584	0.120772	0.977356	0.673673	0.9887	1.839765
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow3.jpg	213.9874	188.3767	71.44858	0.253064	0.977265	0.556545	0.983943	2.466708
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow30.jpg	163.2754	130.4333	24.43884	0.237071	0.627086	0.933246	0.984076	0.331252
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow31.jpg	172.0347	141.2677	25.79864	0.231403	0.602778	0.944929	0.986912	0.262639
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow32.jpg	176.6988	133.1018	13.38692	0.156373	0.60068	0.960709	0.990512	0.195777
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow33.jpg	166.6781	125.5086	12.73923	0.094577	0.975683	0.655552	0.985045	1.925794
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow34.jpg	160.605	114.816	28.192	0.149081	0.544965	0.956694	0.988928	0.207952
C:\Users\HP PC\Desktop\lemonpictures\overmature\yellow35.jpg	184.6241	154.7627	49.80419	0.128738	0.977555	0.669179	0.986356	1.877232

Fig., Color image Features are extracted and saved in the excel sheet

Weka preprocess:-

In preprocess we extract the graph of RMean, GMean, BMean, Contrast, Energy, Homogeneity and class of the color image for the identification of the maturity stage using the color feature.

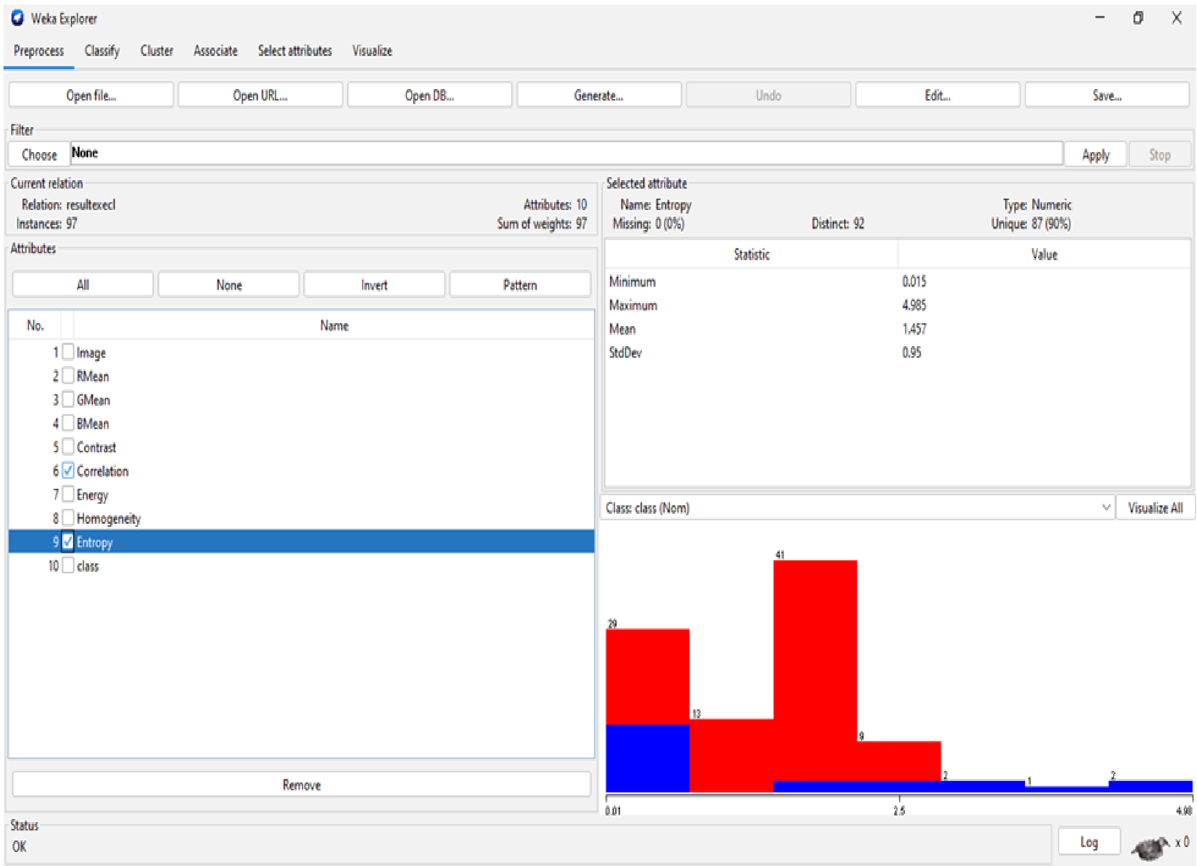

Fig., preprocess Graph

Classifier:

In Weka software we use the classifier to get the accuracy of the given image. Here, we use the function Random Tree classifier to get the accuracy of the matureness of the color image. In the classifier it classifies the output with accuracy of 93.8244% result and it gives the information and summery for the matureness identification of lemon fruit using the color feature.

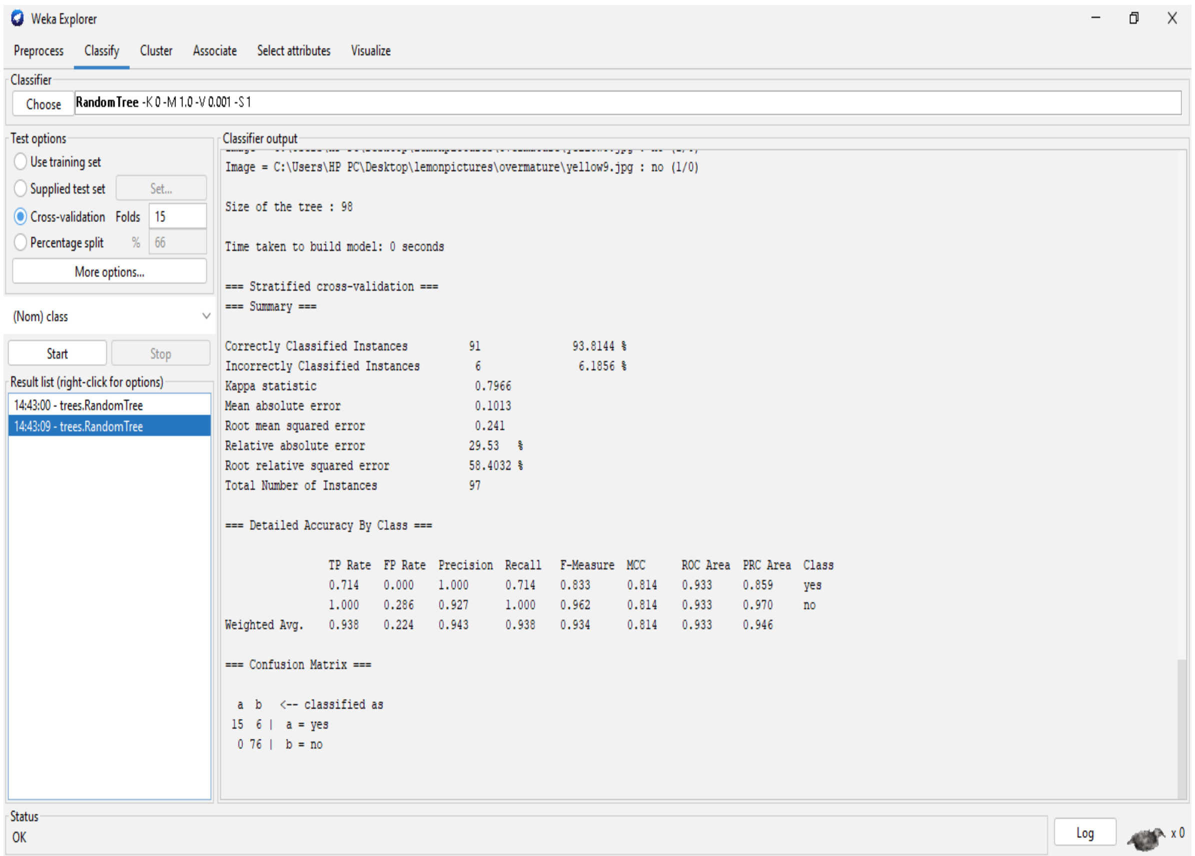

Fig1. Classifier output showing summery classifying matureness using the function Decision Stump classifier.

.

CHAPTER-9

9. CONCLUSION

The developed algorithm is tested with sample data and outputs obtained in according to the requirements. The performance of the system is evaluated, and is found to be much more efficient than the existing system. Though it could not be claimed that ours is an ideal project, it will meet the primary requirements of the concern. Even though we have tried our level best to make it a dream project, due to time constraints we could not add more facilities to it.so the project has to be improved by having modifications as and when the necessity arises in due course.

CHAPTER-10

10. FUTURE ENHANCEMENT

- This algorithm can be developed into an android app, which provides a easy way to implement the concept
- This concept can also applicable for the all the fruit, which changes the color while reaching to the different different maturity stages.
- This concept can be done in automated way, in a single application.

REFERENCES

1. Strano, M.C.; Di Silvestro, S.; Allegra, M.; Russo, G.; Caruso, M. Effect of cold storage on the postharvest quality of different Tarocco sweet orange clonal selections. Sci. Hortic. 2021, 285, 110167. [CrossRe

2. Gonzalez-Molina, E.; R. Dominguez-Perles; D.A. Moreno and C. Garcia-Viguera (2010). Natural bioactive compounds of Citrus lemon for food and health. Journal of Pharmaceutical and Biomedical Analysis 51: 327–345.

3. Khojastehnazhand, M.; M. Omid and A. Tabatabaeefar (2010). Development of a lemon sorting system based on color and size. African J. Plant Sc. 4(4): 122-127. P–159. [CrossRef].

4. Narendra, V. G. and K. S. Hareesh (2010). Prospects of Computer Vision Automated Grading and Sorting Systems in Agricultural and Food Products for Quality Evaluation. International Journal of Computer Applications. 1(4): 174-181.

5. Omid, M.; M. Khojastehnazhand; A. Tabatabaeefar (2010). Estimating volume and mass of citrus fruits by image processing technique. Journal of Food Engineering. 100: 315–321.

6. Elmasry, G.; M. Kamruzzaman; D.W. Sun and P. Alle (2012). Principles and applications of hyperspectral imaging in quality evaluation of agro-food products: A review. Critical Reviews in Food Science and Nutrition. 52(11): 999–1023.

7. Strano, M.C.; Altieri, G.; Admane, N.; Genovese, F.; Di Renzo, G.C. Advance in Citrus Postharvest Management: Diseases, Cold Storage and Quality Evaluation. In Citrus Pathology; IntechOpen: London, UK, 2017; pp. 13

8. Hardy, S. and G. Sanderson (2010). Primefacts for profitable, adaptive and sustainable primary industries, Primefact 980: 1-6.

9. Jafari, A.; A. Fazayeli and M. R. Zarezadeh (2014). Estimation of orange skin thickness based on visual texture coarseness. Department of Agricultural Engineering, Shiraz University, Shiraz 71441-65186, Iran, Biosystems engineering. 117: 73 - 82

10. Pathare. P. B.; U. L. Opara and F. A. Al-Said (2013). Colour Measurement and Analysis in Fresh and Processed Foods: A Review, Food Bioprocess Technol. 6:36–60.

11. Zhang, B.; H. Wenqian; L. Jiangbo; Z. Chunjiang; F. Shuxiang; W. Jitao and L. Chengliang (2014). Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review, Food Research International. 62: 326–343.

BIBLIOGRAPHY

1. En.wikipedia.org
2. www.research.getnet
3. www.mathwoks.com
4. Onlinelibrary.wiley.com
5. Ieeexplore.ieee.org.
6. Hhttps://statweb.stanford.edu/~lpekelis/13_datafest_cart/WekaManual-3-7-8.pdf