JSS MAHAVIDYAPEETHA
SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING
MYSORE-570006
An Autonomous Institute Affiliated to
Visvesvaraya Technological University, Belgaum

"Image Analysis for Apple Surface Defect Detection"

Thesis submitted in partial fulfillment of curriculum prescribed for the award of the degree of
BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

by

Krishnakanth J C
(4JC09CS043)

Sharath P S
(4JC09CS094)

Vinuta Hegde
(4JC09CS122)

Yogeesh R
(4JC09CS124)

Under the Guidance of
Mrs.Vani Ashok
Assistant Professor,
Department of Computer Science & Engineering,
SJCE, Mysore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
May 2013

JSS MAHAVIDYAPEETHA
SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING
MYSORE-570006
Department of Computer Science and Engineering

Certificate

This is to certify that the work entitled **"Image Analysis for Apple Surface Defect Detection"** is a bonafide work carried out by **Krishnakanth J.C(4JC09CS043), Sharath P.S(4JC09CS094), Vinuta Hegde (4JC09CS122),** and **Yogeesh R(4JC09CS124)** in partial fulfilment of the award of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2013. It is certified that all corrections / suggestions indicated during CIE have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Project Guide
Mrs.Vani Ashok
Asst. Professor
Dept. of CS & E
S.J.C.E, Mysore

Head of Department
Dr. C.N.Ravikumar
Professor & Head
Dept. of CS & E
S.J.C.E, Mysore

Place: Mysore
Date:

Examiners: **1**————————

2————————

3————————

DECLARATION

We, **Krishnakanth JC (4JC09CS043), Sharath PS (4JC09CS094), Vinuta Hegde (4JC09CS122)** and **Yogeesh R (4JC09CS124)** students of the Department of Computer Science and Engineering from Sri Jayachamarajendra College of Engineering -Mysore, do hereby declare that this dissertation entitled **"IMAGE ANALYSIS FOR APPLE SURFACE DEFECT DETECTION"** has been independently carried out by us under the guidance of **Mrs Vani Ashok**, Assistant Professor, Dept of Computer Science and Engineering, Sri Jayachamarajendra College Of Engineering, Mysore in partial fulfilment of the requirements for the award of the degree of Bachelor Of Engineering in Computer Science and Engineering, Visvesvaraya Technological University, Belgaum.

We also declare that we have not submitted this dissertation to any other University for the award of any degree.

Krishnakanth J C
Sharath P S
Vinuta Hegde
Yogeesh R

ACKNOWLEDGEMENT

Its an old adage **"Experience is the best teacher"**. The satisfaction and pleasure that accompany the gain of experience would be incomplete without the mention of the people who made it possible.

First and foremost we ought to pay due regards to this renowned institution, which provided us a platform and an opportunity for carrying out this project work.

Our sincere regards to **Dr. B.G. Sangameshwara, Principal of S.J.C.E** for providing us an opportunity to enhance our knowledge by working on this project. We also thank **Dr. C.N. Ravikumar, Professor and Head, Department Of Computer Science and Engineering** for the implementation of plan of the project in our curriculum to enhance the knowledge of the students.

We are very grateful to our Assistant Professor **Smt. Vani Ashok** for her whole hearted encouragement and guidance in carrying out this project.

Our special thanks to Assistant Professor, Miss **Bindu A** for her guidance.

Krishnakantha J C
Sharath P S
Vinuta Hegde
Yogeesh R

Abstract

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory. Computer vision has also been described as the enterprise of automating and integrating a wide range of processes and representations for vision perception.

"IMAGE ANALYSIS FOR APPLE SURFACE DEFECT DETECTION" is a computer vision technology for fruit inspection. This system identifies surface defects on apples, based on analyzing images of apples. Apples with different qualities and damage are used to test the developed algorithm. The defect may be fungi attack, frost damage, bruising, punches, insect holes, and scab. Eight images of each apple were taken. Apple was rotated 45^o between the acquisitions of each frame, a given part of the surface appeared at different positions in as many as eight frames. When multiple images are combined dark areas caused by defects would appear in two or more frames. All images are combined using AND operation, so that final frame can have defects from all images if any. The amount of black area present in final image is going to decide whether the apple is defective or not.

Contents

1

Introduction

Post harvest sorting of apples is a difficult, labour intensive process in the commercial fresh apples industry. In the apple packing industry, fruits are checked either visually by human or sorting machines for quality control before shipping to consumers. In a typical apple packing house, workers are placed along an apple conveyor to inspect the passing apples and remove the ones that are injured, rotten, diseased, bruised or with other defects. After defect inspection, apples are transferred to another line for cleaning, waxing and drying. The final grading lines sort the apples by their size, color, and shape, and then pack them into different boxes based on the grade. After working at the packing line for many hours, even the most experienced workers may feel tired, thus their efficiency may reduce dramatically. Because many apple defects such as bruises, cuts, sooty blotch, and other physical damage cannot be easily found by eyes, finding a reliable, accurate, and efficient apple defect sorting system will be very valuable and has been very challenging. On top of that now a days availability of manual labour is a big problem. As a result manual labour charges are growing higher and higher day by day. This problem leads us to search for an alternative. Fruit inspection is a visual task. So the use of computer vision has attracted much interest. Technology of capturing images has improved a lot. Many inspection works have been replaced by embedded systems, which use computer system and a camera. Image processing technique is base for all this kinds of automatic embedded systems.

Quality of apple fruits depends on size, color, shape and presence-type of defected skin according to the marketing. Visual inspection of apples with respect to size and color by machine vision is already automated in the industry. However, detection of defects is still problematic due to high variance of defect types and natural variability of skin color.

Our project is a collection of simple and cost effective steps. Image Processing takes major role in our project. The images of an apple were taken by rotating it in particular angle, entire apple was covered. Then the images were processed to check whether the apple is defective or not. Shadowless images were taken for processing. Images were re-sized for the purposes of cost effectiveness. We converted RGB images into binary images for further processing. Image thresholding is the key of this project. The amount of black pixels in the final frame decides whether to accept or reject the apple. In short we can say the objective of this project was to develop and test an image processing system which could identify defects on apple surfaces. It was proposed to analyze multiple images acquired while the apples were obtaining in front of the camera.

This is an application of computer vision system. Machine vision systems are largely employed for automatically controlling or analyzing processes or activities in many industries like automotive, electronics, food beverages, pharmaceutical, textile etc... One of the most popular applications of machine vision is to inspect qualities of produced goods based on form, color and presence of defects. Machine vision systems benefit from specially designed image processing softwares to perform such particular tasks, therefore image processing plays a very crucial role in their performance.

1.1 Objective

The objective of this project is to develop and test an image processing system which could identify defects on apple surfaces. A system for identifying surface defects on apples has been designed, based on analyzing images acquired while apples were rotating in front of the camera. When multiple images were combined and adjustments made for rotation, dark areas caused by defects would appear with almost the same shape and at the same place in three or more frames. While minimizing false positives, the classification accuracy was very high. The proposed algorithm was effective in detecting various defects such as bruises, frost damage, and scab.

1.2 Existing Solution

Algorithm to segment patch-like defects on monochrome images were used before. This method could be difficult to apply on bi-color fruits where the defects are darker then the ground color, but lighter than the blush color.

In some systems a Gaussian model was used for color segmentation to segment defects on Golden Delicious apples with two enhancement steps. The detection was effective, but revealed some difficulties. To segment the defects, each pixel of an apple image was compared with a global model of healthy fruits by making use of the Mahalanobis distances. The proposed algorithm was effective in detecting various defects such as bruises, russet, scab, fungi or wounds.

Fourier analysis of apple peripheries is also a good quality inspection technique. This methodology showed the way in which external product features affect the human perception of quality. If the classification involved more product properties and became more complex, the error of human classification increases.

1.3 Proposed Solution

Apples with different surface defects will be selected. The defect may be fungi attack, frost damage, bruising, punches, insect holes, and scab.

Images were acquired using camera. Apple was rotated and eight images of each apple were captured. A filtered image will be produced by converting original image to Gray-scale images. This is a simple threshold segmentation based on flat-field corrected image. These images will be then threshold segmented to identify the defects. All image processing will be done using Multi-Scan with the image processing toolbox. As the apple will be rotated 45^o between the acquisitions of each frame, a given part of the surface appeared at different positions in as many as eight frames. As the apples in this work will be rotated through 360^o, some defects could be visible in more than one frame. To evaluate the performance of the system the defects appeared in more than 2 frames (3 or more than 3) will be considered.

1.4 Scope

- Applicable for unicolor apples like Golden delicious and Granny smith.
- This is a tool to inspect the apples for surface defect detection. Tool flags the defective apples.
- Can be applicable for some other unicolor bulk fruits by changing the threshold value.
- Images should be free from shadow, but reflections are acceptable.
- Image background should be white.

1.5 Time Schedule for Completion of the Project

Time span of the project is four months, from February 2013 to may 2013. Whole project is divided into following stages.

1. Analyzing the problem definition and Literature survey - 1^{st} Feb to 25^{th} Feb
2. Designing solution - March 1^{st} and 2^{nd} week
3. Temporary Data-set collection - March 3^{rd} week
4. Implementation of basic phase - March 3^{rd} week
5. Original Data-set collection - March last week to April 1^{st} week
6. Final implementation and reporting - April 2^{nd} week to May end

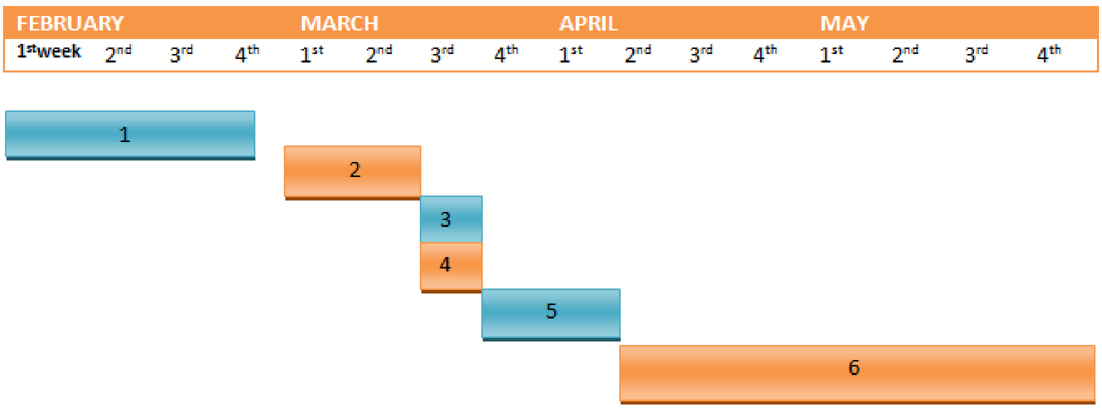

Fig 1.6 : Gantt chart

2

System Requirement and Analysis

2.1 Introduction

Requirement specification encompasses the needs of this project. It aims at providing a full description of the requirements based on the concepts defined in the Problem Domain. The system requirement specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional description, representation of system behaviour, an indication of performance requirements and design constraints, appropriate validation criteria and other information pertinent to requirements.

2.2 Functional Requirements

- The project should be able to flag an apple as defective or not.
- The image processing should be able to identify the area of defect and highlight the same.
- The Image processing program should take 8 frame images from disk and process the same.

2.3 Non-functional Requirements

- The image of apple should be captured in bright light with white background.
- 8 frames of the apple must be captured in each of 360 / 45^o.
- Power should be continuously supplied to all the system involved in this processing without interruption.
- The Apple should of golden delicious of similar kind.
- Try to get an apple image without shadow

2.4 Interface Requirements

- User should be able to select one of the datasets to process.
- Provide dedicated button for user on the click of which the processing of image starts.
- Display the images in dataset chosen, before processing and after processing.
- User should have dedicated button close, on click of which the process stops in between.
- User should get a feel that the image is getting processed.

2.5 Software Requirements

- Operating Systems
  - Windows 8
  - Windows 7 Service Pack 1
  - Windows Vista Service Pack 2
  - Windows XP Service Pack 3

  - Windows XP x64 Edition Service Pack 2
  - Windows Server 2012
  - Windows Server 2008 R2 Service Pack 1
  - Windows Server 2008 Service Pack 2
  - Windows Server 2003 R2 Service Pack 2
- MATLAB 7.8
- Latex
- Ms-Paint
- gedit

2.6 Hardware Requirements

- Processors

  Any Intel or AMD x86 processor supporting SSE2 instruction set
- Disk Space
  - 1 GB for MATLAB only,
  - 3 to 4 GB for a typical installation
- RAM 1024 MB (At least 2048 MB recommended)
- Hardware: Keyboard, Mouse.
- Display: Intel HD graphics (1366X768)
- Camera: Canon EOS 550D (DSLR with flash)

3

Tools and Technology Used

3.1 Image Processing

Image Processing is a technique to enhance raw images received from cameras/sensors placed on satellites, space probes and aircraft or pictures taken in normal day-to-day life for various applications. Methods of Image Processing There are two methods available in Image Processing, analog Image Processing and digital image processing. Various techniques have been developed in image processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecrafts, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics softwares etc. The common steps in image processing are image scanning, storing, enhancing and interpretation. After converting the image into bit information, processing is performed. This processing technique may be, Image enhancement, Image restoration, and Image compression.

The schematic diagram of image scanner-digitizer diagram is shown in figure

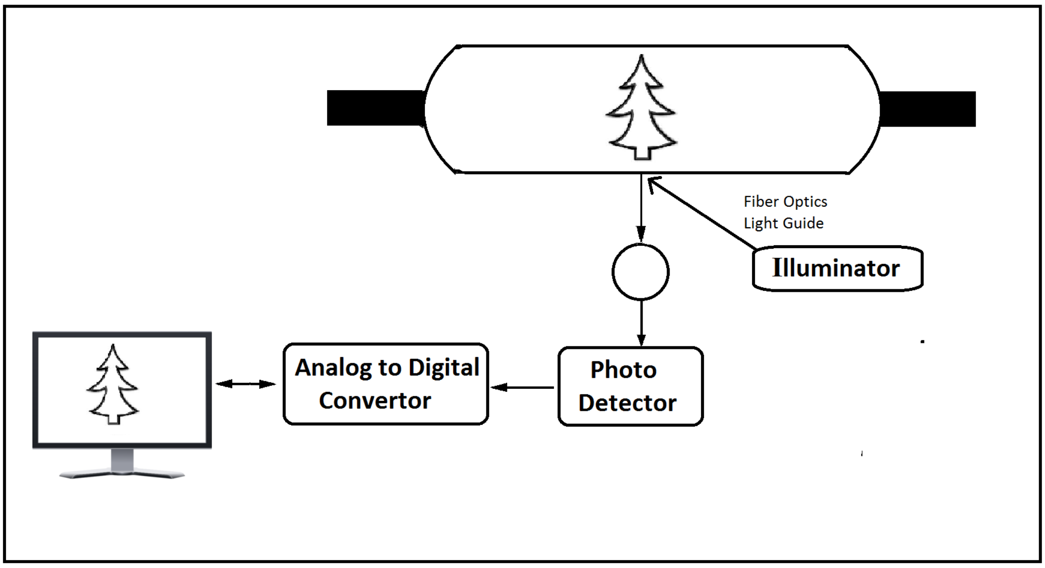

Fig 3.1 : Image scanner digitizer

3.1.1 Images

An image is an array, or a matrix, of square pixels (picture elements1) arranged in columns and rows.

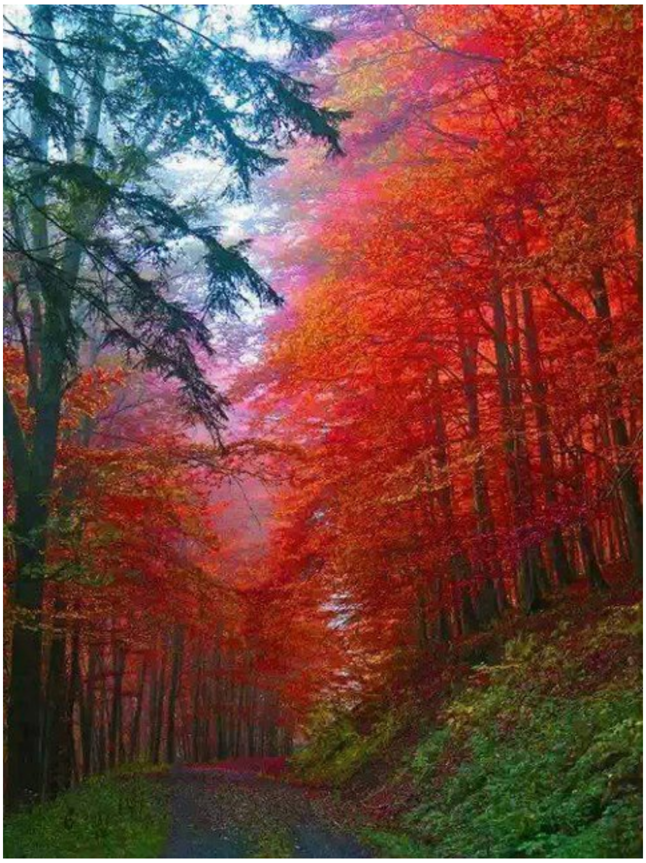

Fig 3.1.1.1 : example image

Image has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. however different computer monitors may use different sized pixels. Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel. Each of these values can be interpreted as an unsigned byte between 0 and 255. Within the color higher numbers are brighter. Thus a red of 0 is no red at all while a red of 255 is a very bright red. Currently Java only supports two levels of transparency: Completely opaque (255) and completely transparent (0). Values of 1 through 254 are treated as completely transparent. Different colors are made by mixing different levels of the three primary colors. For example, medium Gray is 127 red, 127 green, and 127 blue, pure white is 255 red, 255 green, 255 blue.

Binary Images:

Binary images are images whose pixels have only two possible intensity values. They are normally displayed as black and white. Numerically, the two values are often 0 for black, and either 1 or 255 for white. Binary images are often produced by thresholding a grayscale or color image, in order to separate an object in the image from the background. The color of the object (usually white) is referred to as the foreground color. The rest (usually black) is referred to as the background color. However, depending on the image which is to be thresholded, this polarity might be inverted, in which case the object is displayed with 0 and the background is with a non-zero value. Some morphological operators assume a certain polarity of the binary input image so that if we process an image with inverse polarity the operator will have the opposite effect.

Grayscale Images:

A grayscale (or graylevel) image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. In fact a 'gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image. Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the difference between successive graylevels is significantly better than the graylevel resolving power of the human eye. Grayscale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images.

Color Images:

It is possible to construct (almost) all visible colors by combining the three primary colors red, green and blue, because the human eye has only three different color receptors, each of them sensible to one of the three colors. Different combinations in the stimulation of the receptors enable the human eye to distinguish approximately 350000 colors. A RGB color image is a multi-spectral image with one band for each color red, green and blue, thus producing a weighted combination of the three primary colors for each pixel. A full 24-bit color image contains one 8-bit value for each color, thus being able to display $2^{24} = 16777216$ different colors. However, it is computationally expensive and often not necessary to use the full 24- bit image to store the color for each pixel. Therefore, the color for each pixel is often encoded in a single byte, resulting in an 8-bit color image. The process of reducing the color representation from 24-bits to 8-bits, known as color quantization, restricts the number of possible colors to 256. However, there is normally no visible difference between a 24-color image and the same image displayed with 8 bits. An 8-bit color images are based on color maps, which are look-up tables taking the 8-bit pixel value as index and providing an output value for each color.

Color Depth:

Color depth or bit depth is either the number of bits used to indicate the color of a single pixel, in a bitmapped image frame buffer, or the number of bits used for each color components of a pixel. When referring to a pixel the concept can be defined as bits per pixel (bpp), which specifies the number of bits used. When referring to a color component the concept can be defined as bits per color (bpc) or bits per sample (bps). Color depth is only one aspect of color representation, expressing how finely levels of color can be expressed; the other aspect is how broad a range of colors can be expressed (the gamut). The definition of both color precision and gamut is accomplished with a color encoding specification which assigns a digital code value to a location in a color space.

8-bit Color Image:

Full RGB color requires that the intensities of three color components be specified for each and every pixel. It is common for each component, intensity to be stored as an 8-bit integer, and so each pixel requires 24 bits to completely and accurately specify its color. If this is done, then the image is known as a 24-bit color image. However there are two problems with this approach:

– Storing 24 bits for every pixel leads to very large image files that with current technology are cumbersome to store and manipulate. For instance a 24-bit 512*512 image takes up 750KB in uncompressed form.

– Many monitor displays use color maps with 8-bit index numbers, meaning that they can only display 256 different colors at any one time. Thus it is often wasteful to store more than 256 different colors in an image anyway, since it will not be possible to display them all on screen.

Because of this, many image formats (e.g. 8-bit GIF and TIFF) use 8-bit color maps to restrict the maximum number of different colors to 256. Using this method, it is only necessary to store an 8-bit index into the color map for each pixel, rather than the full 24-bit color value. Thus 8-bit image formats consist of two parts: a color map describing what colors are present in the image, and the array of index values for each pixel in the image. When a 24-bit full color image is turned into an 8-bit image, it is usually necessary to throw away some of the colors, a process known as color quantization. This leads to some degradation in image quality, but in practice the observable effect can be quite small, and in any case, such degradation is inevitable if the image output device (e.g.screen or printer) is only capable of displaying 256 colors or less. The use of 8-bit images with color maps does lead to some problems in image processing. First of all, each image has to have its own color map, and there is usually no guarantee that each image will have exactly the same color map. Thus on 8-bit displays it is frequently impossible to correctly display two different color images that have different color maps at the same time. Note that in practice 8-bit images often use reduced size color maps with less than 256 colors in order to avoid this problem. Another problem occurs when the output image from an image processing operation contains different colors to the input image or images. This can occur very easily, as for instance when two color images are added together pixel-by-pixel. Since the output image contains different colors from the input images, it ideally needs a new color map, different from those of the input images, and this involves further color quantization which will degrade the image quality. Hence the resulting output is usually only an approximation of the desired output. Repeated image processing operations will continually degrade the image colors. And of course we still have the problem that it is not possible to display the images simultaneously with each other on the same 8-bit display.

24-bit Color Image:

Full RGB color requires that the intensities of three color components be specified for each and every pixel. It is common for each component, intensity to be stored as an 8-bit integer, and so each pixel requires 24 bits to completely and accurately specify its color. Image formats that store a full 24 bits to describe the color of each and every pixel are therefore known as 24-bit color images. Using 24 bits to encode color information allows different colors to be represented, and this is sufficient to cover the full range of human color perception fairly well. The term 24-bit is also used to describe monitor displays that use 24 bits per pixel in their display memories, and which are hence capable of displaying a full range of colors. There are also some disadvantages to using 24-bit images. Perhaps the main one is that it requires three times as much memory, disk space and processing time to store and manipulate 24-bit color images as compared to 8-bit color images. In addition, there is often not much point in being able to store all those different colors if the final output device (e.g. screen or printer) can only actually produce a fraction of them. Since it is possible to use color maps to produce 8-bit color images that look almost as good, at the time of writing 24-bit displays are relatively little used.

3.1.2 Color Models

For science communication, the two main color spaces are RGB and CMYK.

RGB

The RGB color model relates very closely to the way we perceive color with the r, g and b receptors in our retinas. RGB uses additive colour mixing and is the basic color model used in television or any other medium that projects color with light. It is the basic color model used in computers and for web graphics, but it cannot be used for print production. The secondary colors of RGB cyan, magenta, and yellow are formed by mixing two of the primary colors (red, green or blue) and excluding the third color. Red and green combine to make yellow, green and blue to make cyan, and blue and red form magenta. The combination of red, green, and blue in full intensity makes white. In Photoshop using the screen mode for the different layers in an image will make the intensities mix together according to the additive colour mixing model. This is analogous to stacking slide images on top of each other and shining light through them. The additive model of RGB. Red, green, and blue are the primary stimuli for human colour perception and are the primary additive colours.

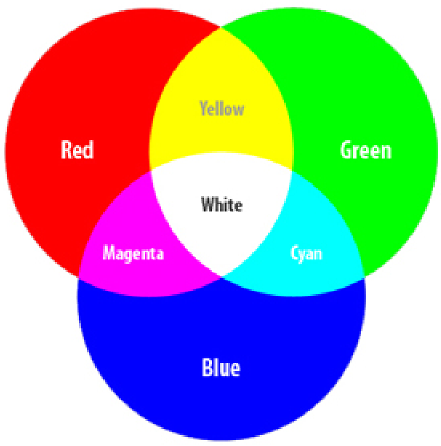

Fig 3.1.2.1 : RGB color model

CMYK

The 4-colour CMYK model used in printing lays down overlapping layers of varying percentages of transparent cyan (C), magenta (M) and yellow (Y) inks. In addition a layer of black (K) ink can be added. The CMYK model uses the subtractive colour model. The colours created by the subtractive model of CMYK don't look exactly like the colours created in the additive model of RGB Most importantly, CMYK cannot reproduce the brightness of RGB colours. In addition, the CMYK gamut is much smaller than the RGB gamut.

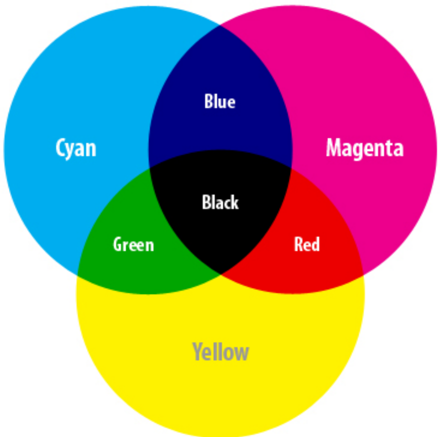

Fig 3.1.2.2 : CYMK color model

Gamut

The range, or gamut, of human colour perception is quite large. The two colour spaces discussed here span only a fraction of the colours we can see. Furthermore the two spaces do not have the same gamut, meaning that converting from one colour space to the other may cause problems for colours in the outer regions of the gamuts. This illustration below clearly shows the different gamuts of the RGB and CMYK colour spaces.

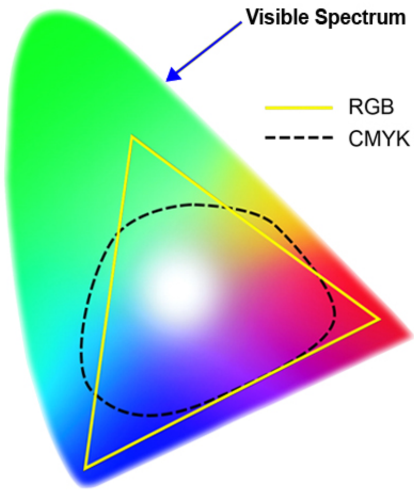

Fig 3.1.2.3 : Whole gamut of human colour perception

3.1.3 Image Processing Operations

This section discusses image processing operations, which are used in this project.

Image Segmentation:
Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels). 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 (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. 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.

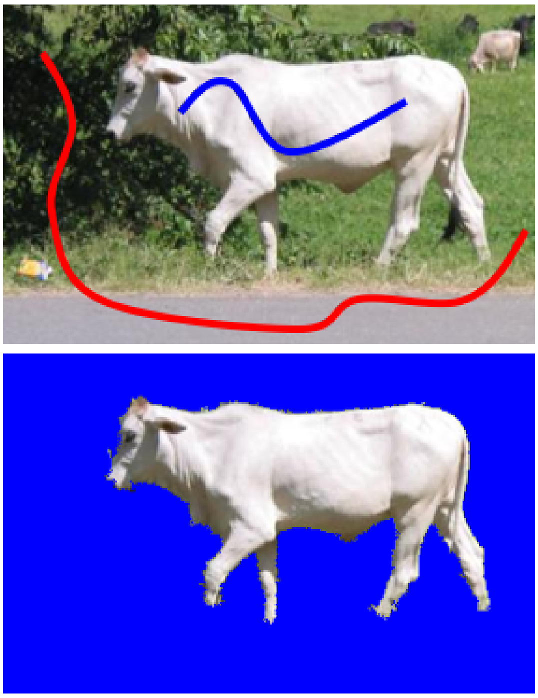

Fig 3.1.3.1 : Image segmention

Image Thresholding:

In image processing, thresholding is used to split an image into smaller segments, or junks, using at least one color or grayscale value to define their boundary. A possible threshold might be 40% gray in a grayscale image: all pixels being darker than 40% gray belong to one segment, and all others to the second segment. Its often the initial step in a sequence of image-processing operations. One key point: These segments are not necessarily convex since image content is arbitrary. In that sense segments should represent objects in the image, e.g. letters, cars, traffic signs, human faces in the foreground. Unfortunately, such objects vary with respect to colors, intensity, illumination, lens aberrations and noise which makes choosing the right threshold not an easy task. Multiple thresholds might be needed in connection with more sophisticated algorithms. In case the object in the foreground has quite different gray levels than the surrounding background, image thresholding is an effective tool for this separation, or segmentation. There are two types in image thresholding:

1)Global thresholding

2)Adaptive, dynamic thresholding.

Global thresholding:

The most straightforward approach is to use one threshold across the whole image. Although its computationally most effective compared to other thresholding techniques, varying illumination/colors and noise strongly influence the quality of the result. In addition, there is no guarantee for contiguous segments; and only intensity values are considered.

Adaptive, dynamic thresholding:

Due to the variability in the gray level intensities and because of noise, the global thresholding does not work satisfactorily. The adaptive dynamic thresholding methods build a threshold surface that is a function on the image domain, and then threshold the image with this threshold surface. There are plenty of methods out there that build this adaptive, dynamic threshold surface.

Noise removal using Image Filters:

Noise is defined as the random graylevel variations within an image that have no spatial dependences from image. This noise can be reduced using image filtering techniques. Image filtering is a process by which we can enhance (or otherwise modify, warp, and mutilate) images. In our case, the type of noise we came across was impulse(salt-and-pepper) noise. The PDF of bipolar noise is given by

$$p(z) = \begin{cases} Pa & \text{if } z=a \\ Pb & \text{if } z=b \\ 0 & \text{otherwise} \end{cases}$$

If b >a, intensity b will appear as a light dot in the image. Conversely, level a will appear like dark dot. if Pa or Pb is zero, the impulse noise is called unipolar. If neither probability is zero, and especially if they are approximately equal, impulse impulse noise value will resemble salt-and-pepper granules randomly distributed over the image. Noise impulses can be negative or positive. Scaling usually is a part of the image digitizing process. Because impulse corruption is usually is a large compared with the strength of the image signal, impulse noise generally is digitized as extreme values of an image. Thus, the assumption usually is a and b are "saturated " values, in the sense that they are equal to minimum or maximum allowed values in the digitized image. As a result, negative impulse appear as black(pepper) points in an image. For the same reason, positive impulse appear as white(salt). Since we wanted to remove small black and/or white spots, we used bwopenarea() function to reduce those noise.

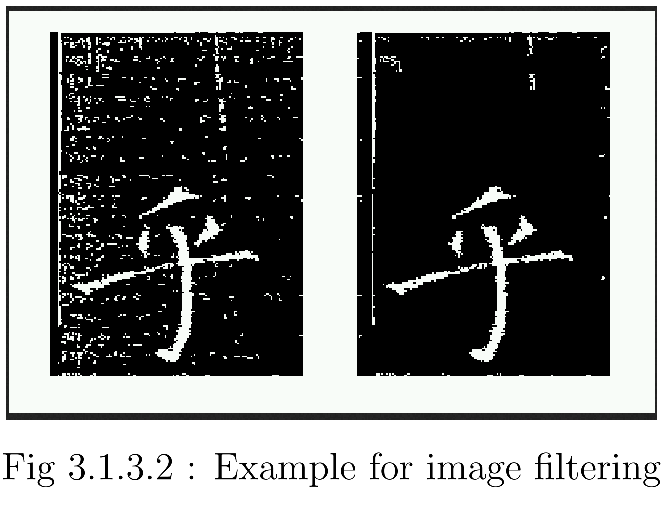

Fig 3.1.3.2 : Example for image filtering

3.2 Tool Used for Implementation

MATLAB

MATLAB(matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems. In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises. We have used as a image processing tool kit and GUI tool kit(Wikipedia). We can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

Matlab is well adapted to numerical experiments since the underlying algorithms for Matlab's builtin functions and supplied m-files are based on the standard libraries LINPACK and EISPACK. Matlab program and script files always have filenames ending with ".m"; the programming language is exceptionally straightforward since almost every data object is assumed to be an array. Graphical output is available to supplement numerical results

3.3 Tools Used for Preparing Report

MS Paint:

MS-Paint is a simple graphics painting program that has been included with all versions of Microsoft Windows. It is often referred to as MS Paint or Microsoft Paint. The program opens and saves files as Windows bitmap, GIF, PNG, and TIFF. The program can be in color mode or two-color black-and-white, but there is no grayscale mode. For its simplicity, it rapidly became one of the most used applications. We have used this tool to edit images. (Wikipedia)

Latex:

LaTeX, is a document mark-up language and document preparation system for the TeX typesetting program. The term LaTeX refers only to the language in which documents are written, not to the editor application used to write those documents. In order to create a document in LaTeX, a. Tex file must be created using some form of text editor. While most text editors can be used to create a LaTeX document, a number of editors have been created specifically for working with LaTeX. LaTeX is widely used in academia. As a primary or intermediate format, e.g., translating DocBook and other XML-based formats to PDF, LaTeX is used because of the high quality of typesetting achievable by TeX. The typesetting system offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout and bibliographies. LaTeX is intended to provide a high-level language that accesses the power of TeX. LaTeX essentially comprises a collection of TeX macros and a program to process LaTeX documents. Because the TeX formatting commands are very low-level, it is usually much simpler for end-users to use LaTeX. We have used this tool to create report. (Wikipedia)

G-editor:
“gedit” is a text editor for the GNOME desktop environment, Mac OS X and Microsoft Windows. Designed as a general purpose text editor, gedit emphasizes simplicity and ease of use. It includes tools for editing source code and structured text such as markup languages. It is designed to have a clean, simple graphical user interface according to the philosophy of the GNOME project, and it is the default text editor for GNOME. We have used this tool to edit text file foe latex. (Wikipedia)

3.4 Tools Used for Collecting Data-set

Camera Canon 550D:

- Type Digital single-lens reflex
- Sensor CMOS APS-C 22.3, 14.9 mm (1.6x conversion factor)
- Maximum resolution 5,184, 3,456 (17.9 recorded megapixels)
- Lens Canon EF lens mount, Canon EF-S lens mount
- Flash E-TTL II automatic built-in pop-up
- Shutter focal-plane
- Shutter speed range 1/4000 to 30 sec and bulb, 1/200 s X-sync
- Exposure metering Full aperture TTL, 63-zone SPC
- Exposure modes Full Auto, Portrait, Landscape, Close-up, Sports, Night Portrait, No Flash, Program AE, Shutter-priority, Aperture-priority,
- Manual, Auto Depth-of-field, Movie
- Metering modes Evaluative, Spot (4% at center), Partial (9% at center), Center-weighted average

- Focus areas 9 AF points, f/5.6 cross-type center (extra sensitivity at f/2.8)
- Focus modes AI Focus, One-Shot, AI Servo, Live View
- Continuous shooting 3.7 frame/s for 34 JPEG or 6 RAW frames
- Viewfinder Eye-level pentamirror SLR, 95% coverage, 0.87 magnification, and electronic (Live View)
- ASA/ISO range ISO 100 to 6400 (expandable to 12800 with Canon Firmware, expandable to 24000 with Magin Lantern firmware) Flash bracketing No
- Custom WB Auto, Daylight, Shade, Cloudy, Tungsten, Fluorescent, Flash, Custom
- WB bracketing 3 stops for 3 frames
- Rear LCD monitor 3 in 3:2 color TFT LCD, 1,040,000 dots
- Storage Secure Digital Card
- Secure Digital High Capacity
- Secure Digital Extended Capacity
- Battery LP-E8 Lithium-Ion rechargeable battery
- Dimensions 129 mm, 98 mm to 62 mm

Halogen lamps

Halogen lamps are used to capture shadowless the images.

Thermocol sheets

Thermocol is used to get even light intensity while capturing the images.

4

Literature Survey

4.1 Introduction

Literature survey chapter provides an overview of previous research on knowledge sharing and intranets. It introduces the framework for the case study that comprises the main focus of the research described in this thesis. It provides a description, summary, and critical evaluation of each work. Survey may include: scholarly journals, books, dissertations, conference proceedings, etc. It may be completed en route to an essay, thesis or dissertation and included in the final project. Or, it Materials may be conducted as its own entity. This was in order to scope out the key data collection requirements for the primary research to be conducted, and it formed part of the emergent research design process. An appreciation of previous work in this area served three further purposes. First, through providing direction in the construction of data collection tools, it guarded against the risk of overload at the primary data collection stages of the project. Second, working the findings from extant literature into a formal review helped maintain throughout the study a sense of the topics perspective. Finally, this activity raised the opportunities for articulating a critical analysis of the actual meaning of the data collected when the data analysis stages of the research were reached.

4.2 Background

Fresh fruits and vegetables provide a variety of health benefits to our daily life. They contain vitamins, minerals, and many elements that help prevent illnesses such as cancer, heart disease, and stroke. A daily diet of fresh produce is highly recommended by health and nutrition authorities. For the past decade, consumption of fresh vegetables and fruits has been increased every year. Consequently, this increase has no doubt raised the public concerns regarding the potential safety and quality issues of the fresh produce. Quality and safety are among the most important criteria for the evaluation of consumable fresh fruits and vegetables. Generally, quality includes external factors such as appearance, texture and flavor. Poor quality of the fresh produce will have a negative effect on people health; thus, the development of effective fruit inspection technologies to ensure the quality of fruits and vegetables is essential for fresh fruits and vegetables market.

The quality of fruits is affected by various factors such as conditions of growing, storage, and handling. In a typical apple processing factory, workers are employed to inspect and remove defective apples, such as those with rots, bruises, injuries, and other defects must be removed at the first stage to prevent cross-contamination and reduce subsequent processing cost. . At the final stage, apples are sorted according to their size, color and shape, and then packed into boxes according to their grades. In some large packing house operations, sorting machines using machine vision technologies are installed to sort apples into different grades based on weight, size, shape, color, defects and other parameters. It is essential for food factory and large market to reduce disease-causing hazards before fresh produce reaches the consumer. In order to guarantee a safe and wholesome product, fruit and vegetable producers are using various methods to improve safety and quality of their products. Since pathogens from fresh products cannot be completely removed using current washing methods, the most effective way to minimize food safety risks is to identify and remove contaminated raw materials from the product stream, prior to processing or fresh-cut preparation, using non-invasive on-line inspection methods that can identify fecal contamination and reduce human errors. Non-destructive inspection methods have been widely used in research to monitor quality and safety attributes of fresh produce. The ability of detecting and classifying fecal contamination and physical damage in fresh produce could highlight produce with a high risk of contamination and alert producers before the product reaches consumers.

4.3 Automated Embedded Systems for Apple Inspection

Activity assessment of fruits using the method of inertia moment and absolute value of diferences: This was introduced by Bragga et al[11]. Biospeckle is an optical technique that was introduced for nondestructive evaluation of activities of biological materials. In the method, coherent laser light illuminates an object of interest. The back scattered light interferes and a speckle pattern is created in an observation plane. If the sample does not show activity, the speckle pattern is stable in time. However in the case of biological samples, the speckle pattern consists of two components: the static one from stationary elements of the tissue and the variable one from moving particles of the tissue. The variable in time speckle pattern is characteristic for biological tissue and has been called as the biospeckle. Bragga et al. (2009) have summarized that processes related with movement of the scattering centers in the tissue, such as cytoplasmic streaming, organelle movement, cell growth and division during fruits maturation and biochemical reactions, are responsible for a certain biospeckle activity. Brownian motions should be considered as a source of biospeckle activity too. The knowledge about biospeckle in relation to fruits and vegetables is still limited. In general, it has been shown that biospeckle activity changes with an age or with some surface properties, for example an infection of a biological object. On the other hand there is lack of consistent biological interpretation of the phenomena. So far, attempts to apply biospeckle methods in biological studies include measurements of blood flow in blood vessels, viability of seeds, activity of parasites in living tissues, analysis of maturation and bruising of fruits and vegetables. These studies showed that decaying of a tissue conditions caused by age, illness/infection or damage, relates with lower biospeckle activity. In the present work, the analysis of the activity in a speckle pattern is conveniently evaluated by a method known as time history of speckle pattern (THSP), which permits the evolution of one column in the speckle pattern during the time of observation. The numerical analysis of this THSP is carried out by means of a co-occurrence matrix (COM) that assembles the intensity distributions of a speckle pattern with regard to time. Experiments have been performed to implement inertia moment (IM) method on the biospeckle COMs which generates reliable information on the bio-activity of living tissue of the fruits. Also one of the most recently developed method known as absolute value of the differences (AVD) is being analyzed by handling it on biospeckle COM. For the study of biospeckle temporal properties of the fruits, a 2 mW He-Ne laser at 632.8 nm wavelength were expanded and spatially filtered and then is allowed to fall on the specimen. The diffused reflected light from the specimen forms a speckle field in space. The speckle intensity is recorded by a CCD camera with a lens and so the subjective speckle field is recorded. For the experiment apple and tomato were taken as fresh as possible and stored in a cool place. The observation area was marked on each of them. 225 images were taken at an interval of 1 sec to find the IM and AVD values against its activity values. Also 200 images were taken at an interval of 1min to find the variation of IM and AVD value against time. Successive images were registered, digitized to 8 bits (256 grey levels) and stored. Very low illuminating intensity was used to minimize the effect of the irradiation on the sample activity. Measurements were performed on the same places of the samples. For the co-occurrence matrix analysis, a column of the free propagation speckle pattern was read every 1sec for the first result and every 1 min for the second result and then, a composite image of 225 by 225 pixels was formed by stacking consecutive columns. Finally, this image was retrieved and the second-order moments of its co-occurrence matrix were calculated. The speckle images were then registered, the THSP was constructed and the IM was calculated. The whole set of reading was repeated three times and for each of them IM and AVD were calculated. Two types of Indian fruits namely apple and tomato (treated as fruit) have been selected. We have taken a series of observations for the fruits and repeated these observations three times almost in identical situation with temperature in the range of 20 -25^o C and humidity in the range of 55- 60^o. The figure below shows the co-occurrence matrices of apple and tomato for different days respectively for which we can consider it to have high, intermediate and low activities respectively. It can be seen that the points in the main diagonal represent no change of intensity while the spread of points out of the diagonal represents time intensity changes. So, if the activity of the biological tissue is low, intensity changes are slow and the only appreciable values of the matrix appear near to the diagonal .Conversely, if activity is high, the fast intensity changes produce high values far away the principal diagonal of the matrix. Hence it is found that the activity goes on decreasing with time.

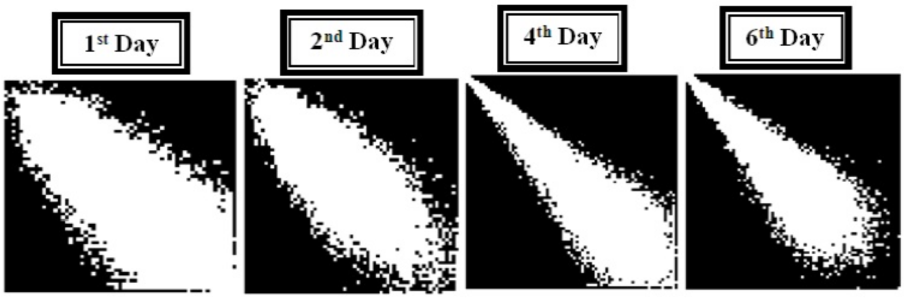

Fig 4.3.1.1 : Co-occurrence matrix MCO for 1st, 2nd, 4th and 6th day of Apple

Gabor feature-based apple quality inspection using kernel Principal component analysis: This method introduces a Gabor feature-based kernel principal component analysis (PCA) method by combining Gabor wavelet representation of apple images and the kernel PCA method for apple quality inspection using near-infrared (NIR) imaging[12]. First, Gabor wavelet decomposition of whole apple NIR images was employed to extract appropriate Gabor features. Then, the kernel PCA method with polynomial kernels was applied in the Gabor feature space to handle non-linear separable features. The results show the effectiveness of the Gabor-based kernel PCA method in terms of its absolute performance and comparative performance compared to the PCA, kernel PCA with polynomial kernels, Gabor-based PCA and the support vector machine methods. Using the proposed Gabor kernel PCA eliminated the need for local feature segmentation, but also resolved the non-linear separable problem. An overall 90.6% recognition rate was achieved. The machine vision system for apple quality inspection consists of a computer-controlled image grabbing module and a NIR sensing system, which is a monochromatic CCD camera with a C-mount lens of 16 mm focal length and a 700 nm long-pass interference filter. The NIR images are grabbed and analyzed by a host computer equipped with a Matrix Meteor/RGB imaging board.

A lighting chamber was designed to provide uniform illumination for the infrared sensor. The 120(W) 100(L) 25(H) cm3 chamber is made of lattice-patterned sheet metal, and the v-shaped interior surface of the chamber is painted flat white to provide diffuse light reflection and eliminate shadows. In order to provide lighting, ten warm-white fluorescent lamps are mounted uniformly around a v-shaped surface right above the conveyor. The NIR imaging sensor is mounted inside on the top center of the chamber. A roller conveyor belt is built to hold and move apples in up to six lanes. All apple samples are manually placed on the conveyer belt with a random orientation. The apples are rotating and moving when they pass through the field of view of the NIR camera. The whole surface of each apple can be covered by the NIR camera during the apple rotation. A drive controller and speed controller are connected with an optical encoder that provides precise timing signals for both on-line mechanical and electrical synchronization.

Fig 4.3.2.1 : NIR Machine Vision System for Apple Quality Inspection.

Total of 40 NIR apple images were used as training samples, and 126 apple images were tested in this research. The dataset was divided into two categories: good or blemished. The proposed Gabor-KPCA method had the highest recognition rate comparing to other approaches. Note that the recognition rates of Gabor PCA and kernel PCA were lower than PCA, while the combination of them was higher. This shows that more information can be reserved through linear representation in original NIR image space rather than in the Gabor feature space.

Apple Defect Detection using Hyperspectral Analysis: Hyperspectral imaging is a non-destructive detection technology and a powerful analytical tool that integrates conventional imaging and spectroscopy to get both spatial and spectral information from the objects for food safety and quality analysis. A recently developed hyperspectral imaging system was used to investigate the wavelength between 530nm and 835nm to detect defects on Red Delicious apples. The combination of band ratio method and relative intensity method were developed in this paper[13], which using the multispectral wavebands selected from hyperspectral images. The results showed that the hyperspectral imaging system with the properly developed multispectral method could generally identify 95% of the defects on apple surface accurately. The developed algorithms could help enhance food safety and protect public health while reducing human error and labor cost for food industry. The apples used in this study were 169 Red delicious apples, randomly selected. Among them, 89 of them are normal good apples, and the rest 80 apples have visible defects. Wax coating was applied to the apple post-harvest. They are stored in a 4^{o}C cold room in the Environmental Microbiological and Food safety laboratory. Beltsville Area research Center, Agricultural Research Service, United States Department of Agriculture. We use the aforementioned hyperspectral line-scan imaging system to inspect each of the 169 apples. After the line-scan image data is obtained, we randomly split the data into two groups: the calibration group and the testing group. The calibration group randomly selected10 normal apples and 15 defect apples, the remaining 79 normal apples and 65 defect apples were in the testing group. The hyperspectral image data in the calibration group is used to determine the optimal wavebands to generate the multispectral imaging algorithm based on the selected optimal wavebands. For the 10 normal samples, we randomly select 2817 pixels from each data. For the 15 apples with defects, we randomly select 3901 pixels from the normal skin area and 794 pixels from the defect area skin for evaluation. The apples were first inspected by relative intensity method, if the apple is detected as good apple, then it is the good apple; if the apple is detected as defect apple, then it will be further inspected by band ratio R1, if the apple is detected as defect one using band ratio R1, then it will be sent using band ratio R2 for further inspection, otherwise, the apple will be recognized as the good one. To test the proposed combined classification method has a better classification rate, I tested three other combined methods, they were band ratio R577/643 and R657/676 using logic AND operation and logic OR operation; band ratio with logic OR operation then combined with relative intensity method using logic OR operation. The four combined methods describe as follows:

B1= R1 AND R2

B2= R1 OR R2

B3= I AND (R1 OR R2)

B4= I AND R1 AND R2

Where B1, B2, B3, B4 are binary image results of four combined detection methods, R1 is the binary image ratio of R577/643 (R577/643 = I577/I643) with threshold T1, R2 is the binary image of R657/676 (R657/676 = I657/I676) with threshold T2, I is the relative intensity of the apple image at 779nm. To get a binary image result, different thresholds are applied to band ratio and relative intensity method. In this research, we define T1 as the threshold value for R577/643, T2 as the threshold value for R657/676 and T3 is the threshold value for I799, respectively. To successful recognize a normal apple, all pixels in a normal apple should have R577/643 value less than T1, R657/676 and I779 value large than T2 and T3 respectively. For successful detection of defect area, at least one pixel in defect apples should have 28 R577/643 value higher than T1, R657/676 and I779 value should lower than T2 and T3 respectively.

Using Statistical Histogram Based EM Algorithm for Apple Defect Detection: Segmentation of an image into its components plays an important role in most of the image processing applications. In this article an important application of image processing in determination of apple quality is studied, and anautomatic algorithm is proposed in order to determine apples skin color defects. First, this image is converted from RGB to color space L*a*b*. Then fruit shape is extracted by ACM algorithm. Finally, the image has segmented using SHEM algorithm. Experimental results on the acquired images show that both EM and SHEM spend the same iterations to accomplish the segmentation process and get the same results. However, the proposed SHEM algorithm consumes less time than the standard EM algorithm. Accuracy of the proposed algorithm on the acquired images is 91.72% and 94.86% for healthy pixels and defected ones, respectively. In this paper, the proposed method has only been evaluated on green and yellow apples[16]. In this paper, an important application of image processing in determination of apple quality is mentioned and an automatic algorithm is presented in order to determine apples skin color defects. The algorithm consists of three stages: First of all, the image was converted from RGB to color space L*a*b*. Secondly, fruit shape was extracted using ACM algorithm. At last, in the third stage, the image is segmented by using SHEM algorithm. Experimental results on the apples data set indicate that both EM and SHEM spend same iterations to accomplish the segmentation process and obtain the same results. But the proposed SHEM algorithm consumes less time than the standard EM algorithm. Accuracy of the proposed algorithm on the acquired was 91.72% and 94.86% for healthy pixels and defected ones, respectively. The methods mentioned in introduction are of type supervised and have been applied on datasets which we do nothave access to. Moreover, our proposed method is of type unsupervised. We have compared the proposed method with EM-based ones leading to similar results. However the proposed method (SHEM) outperforms the EM-based algorithms faster.

Detection of bruises on apple using neae-infrared hyperspectral imaging: The objective of this research was to investigate the potential of near-infrared (NIR) hyperspectral imaging for detecting bruises on apples in the spectral region between 900 nm and 1700 nm. An NIR hyperspectral imaging system was developed and a computer algorithm was created to detect both new and old bruises on apples. Experiments were conducted to acquire hyperspectral images from Red Delicious and Golden Delicious apples over a period of 47 days after bruising. Results showed that the spectral region between 1000 nm and 1340 nm was most appropriate for bruise detection. Bruise features changed over time from lower reflectance to higher reflectance, and the rate of the change varied with fruit and variety. Using both principal component and minimum noise fraction transforms, the system was able to detect both new and old bruises, with a correct detection rate from 62% to 88% for Red Delicious and from 59% to 94% for Golden Delicious. The optimal spectral resolution for bruise detection was between 8.6 nm and 17.3 nm, with the corresponding number of spectral bands between 40 and 20. This research shows that NIR hyperspectral imaging is useful for detecting apple bruises. With improvement in image acquisition speed and detector technology, the NIR hyperspectral imaging technique will have the potential for offline inspection and online sorting of fruit for defects. [17]

5

System Design

5.1 Introduction

System design describes any constraints in the system design and includes any assumptions made by the project team in developing the system design. It is the process or art of defining the hardware and software architecture, components, modules, interfaces, and data for a computer system to satisfy specified requirements. One could see it as the application of systems theory to computing. The design of the system is essentially a blueprint, or a plan for a solution for the system. We consider a system to be set of components with the clear and defined behaviour, which interacts each other in a fixed, defined manner, to produce some behaviour or services to its environment.

Our system is a software, which can be used in apple inspection embedded systems. Image frames of each apple are captured and produced as an input to the system. It flags defective apples, to separate them from heap of apple.

5.2 Data Flow Diagram

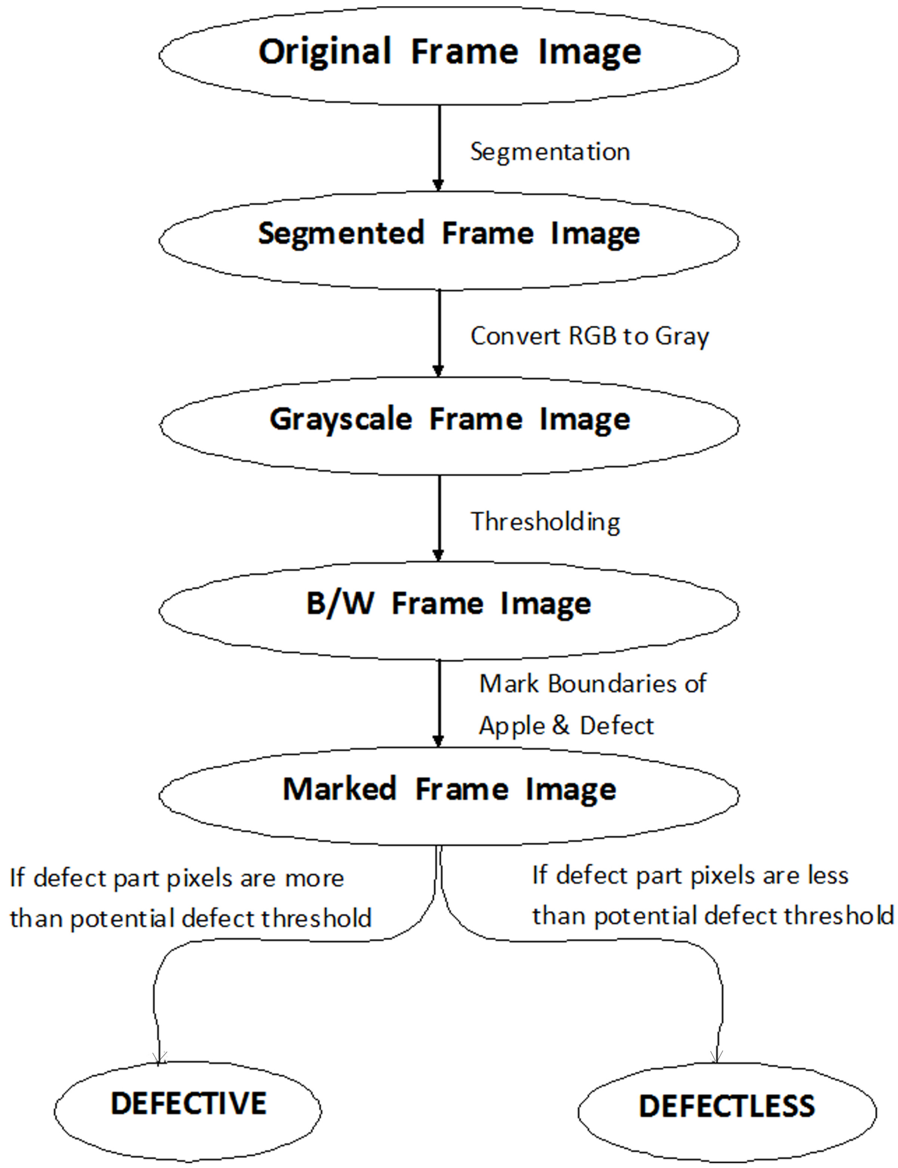

Fig : 5.2.1

5.3 Algorithm

- Get the input from the disk(i.e., eight frames of the apple taken by rotating 45^0 each time which covers the whole surface of apple)
- For each frame of the apple,
  * Process the image by segmenting the apple (i.e., remove the background from the image and taking only the region of in-

terest, Apple)

  * Convert the segmented apple image to grayscale.
  * Binarize the resultant image so that all pixels are either 0 or 1.
  * Thresholding is done for the image to particular value so that the surface defect in the apple gets segmented.
  * Mark the boundaries of apple and the defect to verify that the defect is inside the apple or out of the apple boundary.

- Considering defects in all the frames of the apple, flag it either as a defective apple or non defective one.

5.4 Detailed Design

5.4.1 Data Design

Input for our system is group of image frames. To analyze an apple, 8 frames of each apple are needed. Apple should be rotated 45^{o} before capturing each frame. Background of images should be white. Light intensity should be equal all over the apple. Image frames should be free from shadow. All eight frames should be of same size.

5.4.2 Logical Design

Frame processing

Frame processing is phase in which captured images are converted in to binary image frames, which can be directly used in next stage. After collecting eight images of each apple, by turning the apple 45textsuperscripto between each capturing, they are renamed with integers in increasing order. This makes image frame access easy. The sequence in which the images are captured should be maintained throughout the process. Each frame is extracted from corresponding dataset sequentially. Frames are processed individually. Input images should be free from shadow. But reflection is acceptable. Apple is separated from the back ground. Green component is taken as parameter and apple is separated using image segmentation technique. Then the image frame is converted in to Gray-scale image. In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values. A simple segmentation technique that is very useful for scenes with solid objects resting on a contrasting background. All pixels above a determined (threshold) grey level are assumed to belong to the object, and all pixels below that level are assumed to be outside the object. The selection of the threshold level is very important; as it will affect any measurements of parameters concerning the object. Histograms are used to select the threshold level.

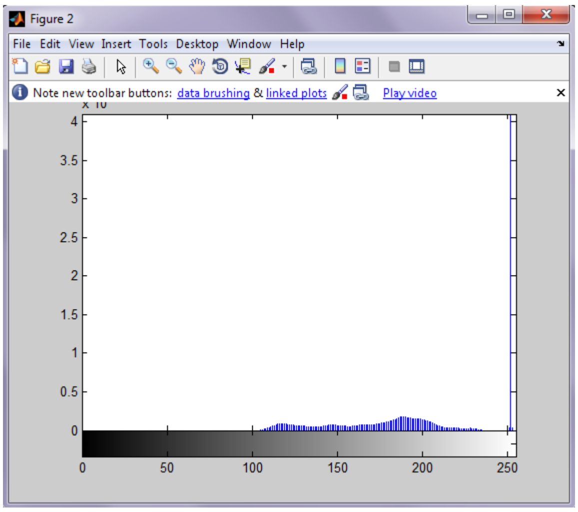

Fig 5.4.2.1 : Histogram for a defect less apple image

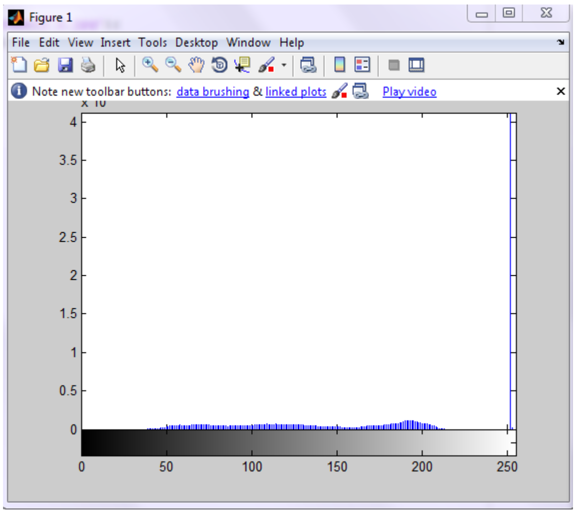

Fig 5.4.2.2 : Histogram for a defected apple image

That Gray-scale image is thresholded using Gray to binary converting function available in MATLAB with threshold parameter value 0.35. Images are thresholded to extract the defects from the apple. This thresholing converts all the pixels which has intensity level more than 0.35 to white and other pixels to black. Small unwanted dots inside the defects are removed to round off the defect shape. Shape of the apple is highlighted in red color and shape of the defects(if any defects are present) are highlighted in green color. All eight images are processed in the same way one by one. Processed images are stored in particular directory. At this stage frames are ready for next phase.

Concluding result using processed frames

All processed image frames in output directory are combined using binary AND operation, by taking 2 image frames at a time. Each pixel in one frame is ANDed with corresponding pixel in the second frame. As we know, AND operation gives high output only when both values are high. Since we need resultant frame in such a way that, if either one of the two pixel is black (low), the resultant pixel should also be black. The resultant pixel will be white (high) only when pixels from both images are white. This requirement is same as binary AND operation. Hence AND operation is used here. After

ANDing all eight images the resultant frame is checked to conclude the result. Numbers of Black pixels in final image are calculated. If number of black pixels in final image exceeds threshold valve, the processed apple is marked as defective, else the apple is considered as non defective.

5.4.3 Interface Design

Since it is a software for an automated system, interface requirements are less compared to other types of system. Our system provides an option to select the dataset for the demonstration purpose. When we select the dataset, the name of the breed of apple will be displayed. It also provides termination button. When the inspection process in done, the result will be displayed in a small pop out window.

6

System Implementation

6.1 Data Collection

The sample apple Granny Smith was taken fresh at the local super market. This being the first step of data collection involved in selecting suitable scar-less sample which would be representing the fresh and healthy apple without any unevenness. The second step being the image capture was done against white background and light of uniform intensity. The uniform white light was made to fall on the apple and apple was rotated 45 degree each time to have 8 pictures of the same apple. Suitable sample with the scar and defect was also collected from the super market and the above mentioned steps were repeated for the same.

Challenges met: The first trial of data collection met with the problem of shadow and bad light in the image which resulted in the fl-awful detection of the defect. Shadow and bad light were the major obstacles we faced during the data collection.

Second trial: For suitable lighting arrangement lighting was provided with the help of reading lamps but this trial was unsuccessful because of unevenness in the focusing of light due to which the glares on the apple will be reflected and when analyzed the glare will be detected as defect.

Third trial: The defective sample was having the defect completely (The apple was rotten). The thresholding level was changed again. All these obstacles were cleared and these were taken as the opportunities to make our project error free still.

Final trial: White uniform sunlight was distributed using white thermocol and the thresholding level was programmed such a way that irrespective of the nature of the defect image will be analyze.

6.2 Interface Implementation

Graphical user interface (GUI) is a type of user interface that allows user to interact easily using graphics rather than text commands. A GUI represents the information and actions available to a user through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. The actions are usually performed through direct manipulation of the graphical elements. A GUI allows users to perform tasks interactively through controls such as buttons and sliders. Within MATLAB, GUI tools enables us to perform tasks such as creating and customizing plots (plottools), fitting curves and surfaces (cftool), and analyzing and filtering signals (sptool). We can also create custom GUIs for others to use either by running them in MATLAB or as standalone applications.

In our implementation, we have used the user interface to interactively take the input from the user, the dataset which has to be processed and flag whether the apple is defective or non-defective on.

We have also given the user, an option to interrupt during the processing of apple dataset. GUI contains the list of datasets which has to be processed, textbox to show what dataset has been chosen by the user, RUN button to run the chosen dataset and finally CLOSE button to interrupt the process and close it.

1. Module showtitle( ) is for displaying the type of fruit you have chosen before running it for the results.
2. Module run( ) is the function where the actual logic for finding the defect is implemented. In this, the dataset which we have chosen is taken as input and processed to display the results i.e., flags the given dataset as a defective fruit or non-defective one.

6.3 Software Implementation

This section narrates step by step step by step implementation and control flow of our system.

- First step is to extract images from the disk.
  * Extracted name of directory selected by the user, in a variable call 'dirname'. Then made it as a image source.
  * Extract i^{th} image and make it as a current file and store the file name in a variable 'filename'.
  * Read image from 'filename' and put it in a variable 'I'.

- Second step is to segment the image to extract the apple from the image frame.
  * By observing a green component distribution in an image, the pixels which has got value of green component in between 65 to 130 are stored in a variable 'kf'.

* The variable 'kf' contains a white apple with black background. The small dots inside the white apple will be removed with the help of imfill() and strel() functions.

- Take an image I and convert it in to grayscale image using rgb2gray() function and store it in IG.
  * Eliminate hue component.
  * Eliminate saturation component.
  * Retains only luminance component and converts RGB color maps to grayscale.

- Segment the image using thresholding technique. Convert IG to binary image by specifying 0.35 as threshold value using im2bw() function.
  * Check the luminance level.
  * Replaces all pixels in the input image with luminance greater than 0.35 with the value 1 (white).
  * Replaces all pixels in the input image with luminance lesser than or equal to 0.35 with the value 0 (black)

- Get rid of unwanted noise pixels using bwareaopen() function.
  * Invert binary image IG.
  * Remove all the objects with size lesser than 40 pixels.
  * Invert an image again to get required image.

- Combine white apple image from and IG using & operation.

- highlight the apple boundary with red color and defect in green color.
  * Store boundary of objects using bwboundaries() function.

    * Plot the boundary with the help of plot() function.

– After repeating all steps for all 8 images, store them in output directory.
– Display all 8 input images in single frame using subplot() function.
– Display all 8 output images in single frame using subplot() function.
– Read all 8 output images and store them in different variables.
– Combine two successive images at a time using & operation. Repeat combining until we left with single frame.
– Count the number of black pixels in the final frame
– Flag the apple as defective only if number of black pixels exceeds the threshold level.

7

Testing and Results

7.1 Introduction

Testing is intended to show that a system conforms to its specification and that the system meets the expectation of the user of the system. Large systems are built out of the sub systems which are built out of modules which are composed of procedures and functions. The testing process should therefore proceed in stages where testing is carried out incrementally in conjunction with system implementation. The testing of the learning tool was done along with the implementation of the various modules. This method of testing helps to ensure the proper working of the modules at the time of their implementation. Testing involves exercising the real data processed by the program. The existence of program defects or inadequacies is inferred from unexpected system outputs. For verification and validation, we make use of the program testing technique.

7.2 Unit Testing

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect.

Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use. This type of testing is driven by the architecture and implementation teams. This focus is also called black-box testing because only the details of the interface are visible to the test. Limits that are global to a unit are tested here.

Image segmentation using color component

Fig 7.2.1 : Input image

Fig 7.2.2 : Segmented image

RGB to Gray-scale conversion

Fig 7.2.3 : Input image

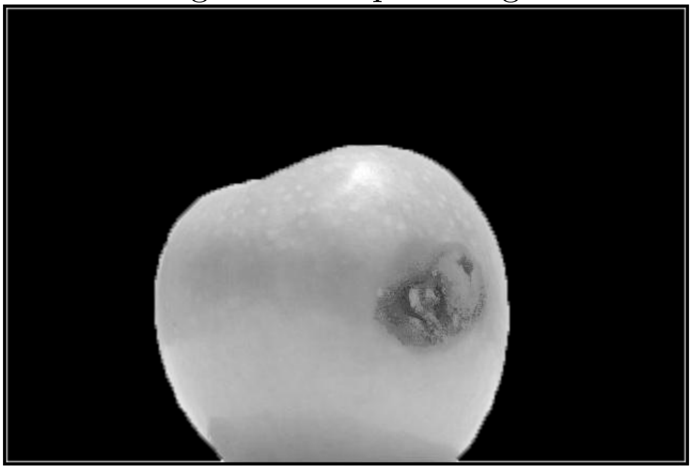

Fig 7.2.4 : Grayscale image

Image segmentation using thresholding

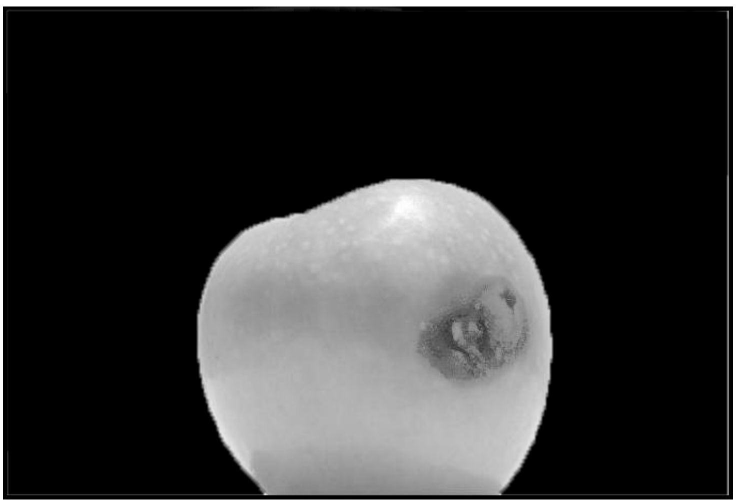

Fig 7.2.5 : Input image

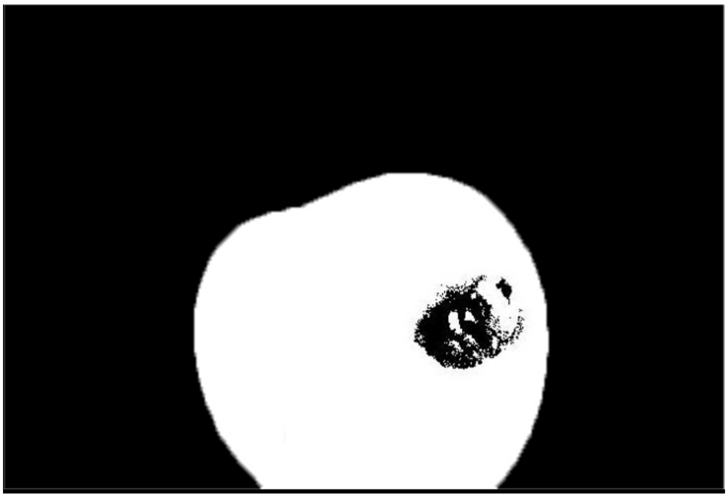

Fig 7.2.6 : Thresholded image

Noise reduction

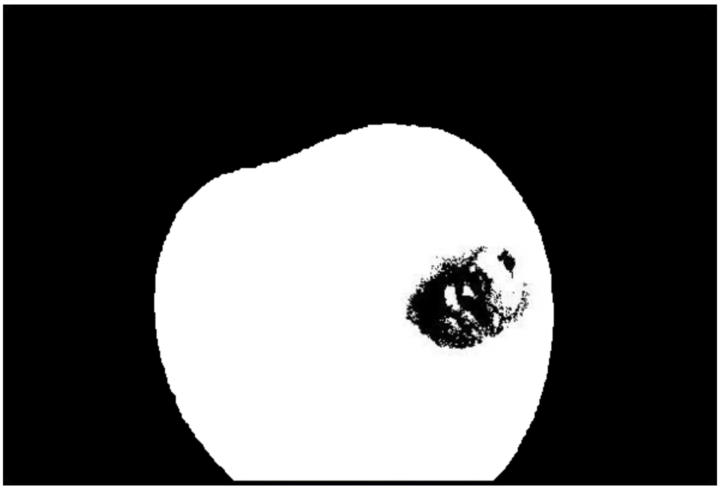

Fig 7.2.7 : Input image

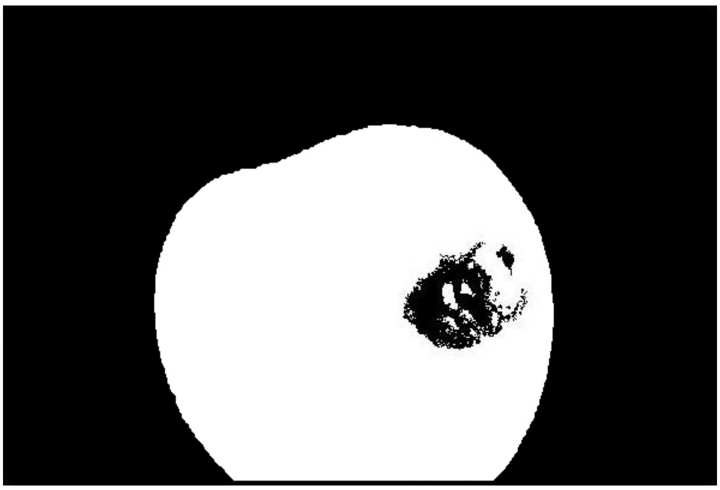

Fig 7.2.8 : Image after noise reduction

7.3 Experimental Results

Experiment on dataset with defects:

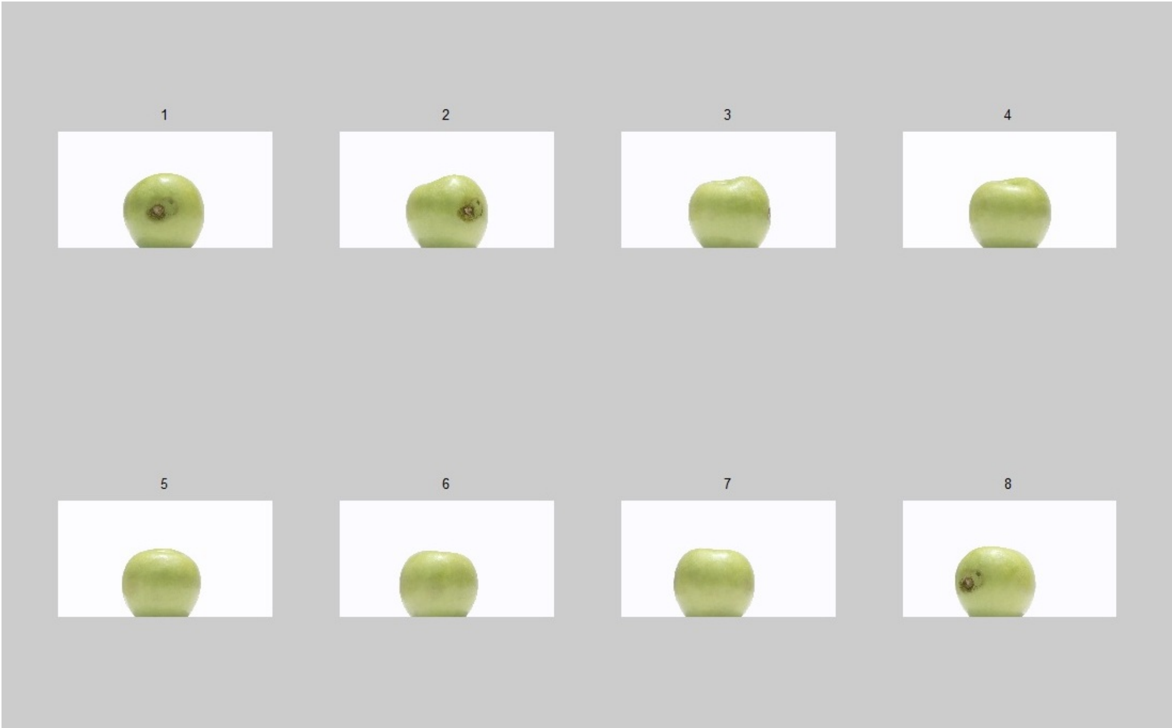

Fig 7.3.1 : Input image frames with defects

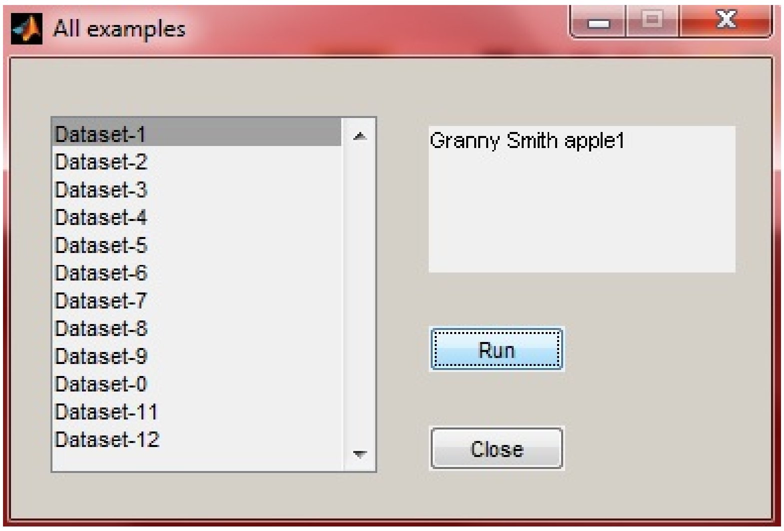

Fig 7.3.2 : Select window displaying name of the selected dataset

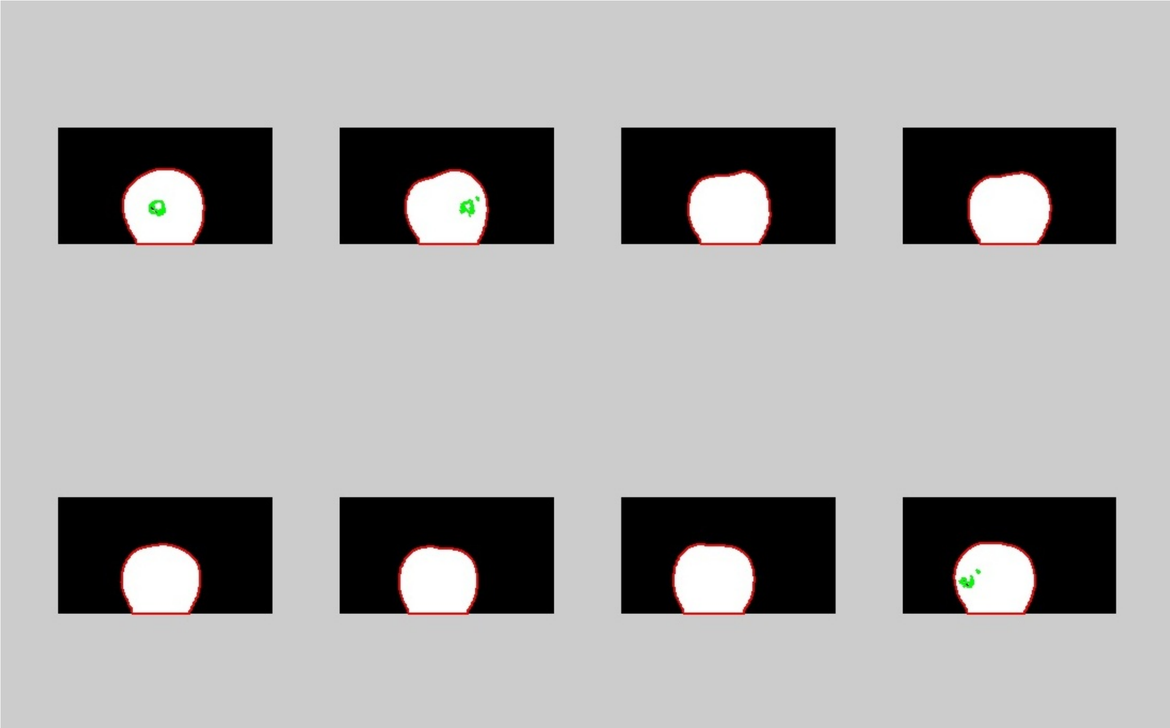

Fig 7.3.3 : Images after frame processing

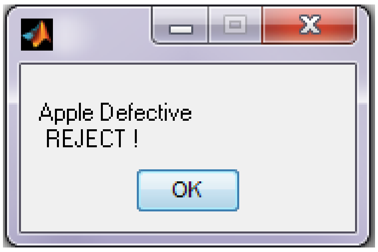

Fig 7.3.4 : Pop up window displaying result

Experiment on dataset without defects:

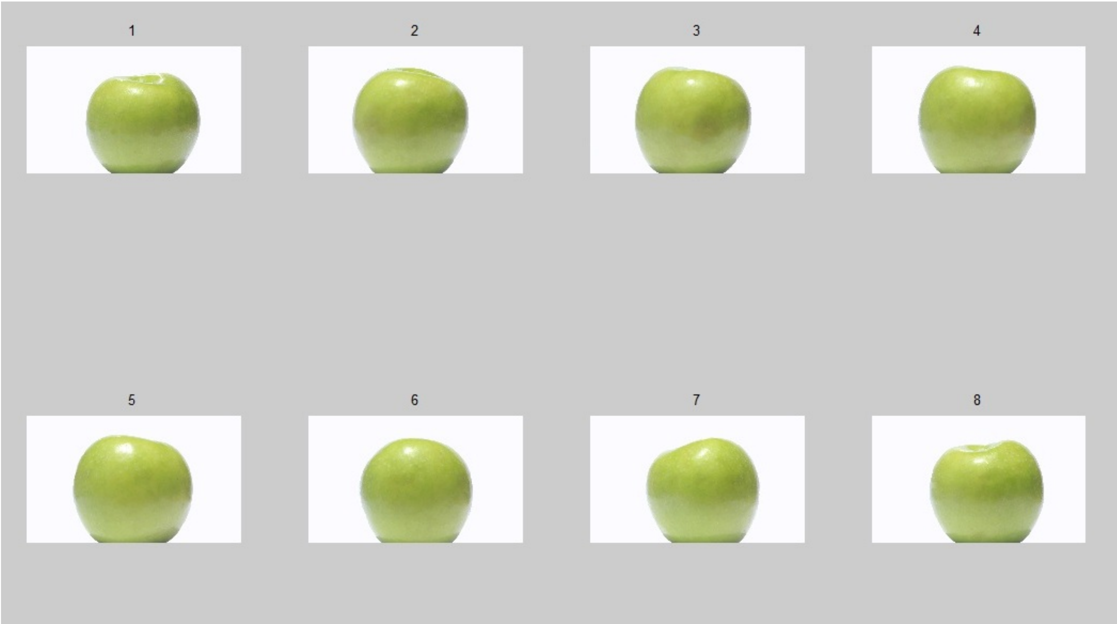

Fig 7.3.5 : Input image frames without defects

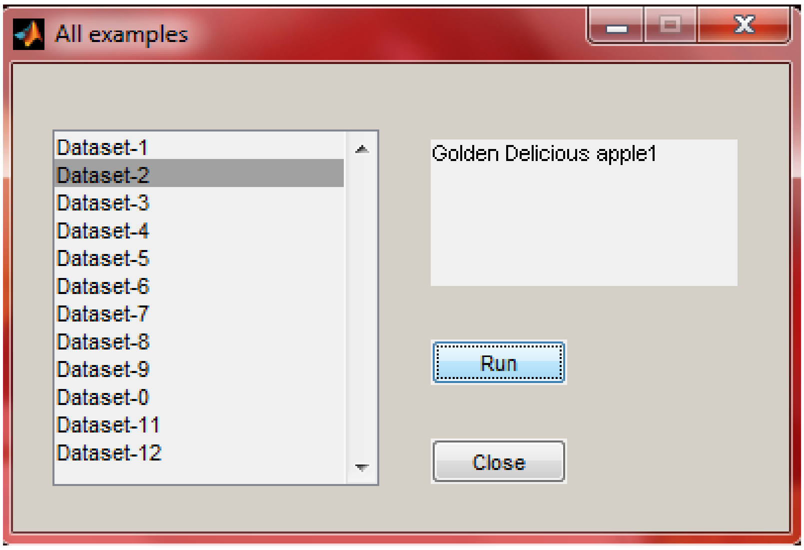

Fig 7.3.6 : Select window displaying name of the selected dataset

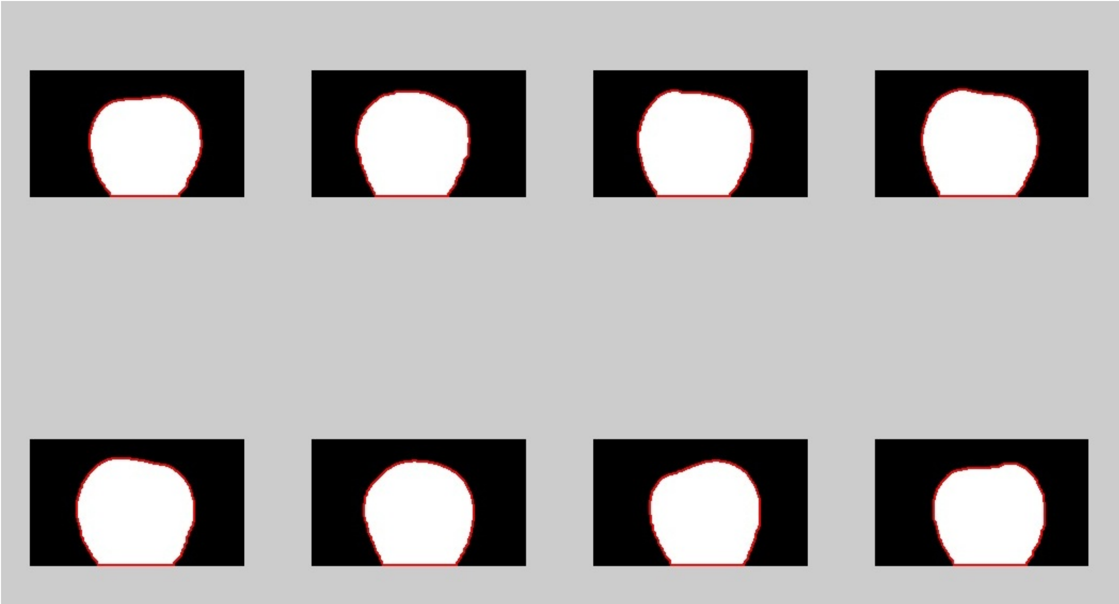

Fig 7.3.7 : Images after frame processing

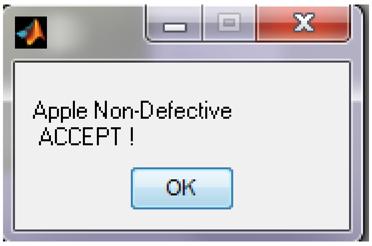

Fig 7.3.8 : Pop up window displaying result

Table 7.1: Resultset(GUI)

Sl no	Input description	Expected result	Actual result	Status
1	Select valid dataset and click run	Combined input & output frames with result display	Combined input & output frames with result display	Pass
2	Select invalid dataset and click run	No output	No output	Pass
3	Click run without selecting any dataset	No output	No output	Pass

Table 7.2: Resultset(Entire system)

Sl no	Input description	Expected result	Actual result	Status
1	Granny smith with defect	Defective	Defective	Pass
2	Granny smith without defect	Not defective	Not defective	Pass

8

Application

The "Image analysis for apple surface defect detection" project is boon for defect analysis in the shipping industry and apple farms. The legacy system which neglects some specific breeds like golden delicious and granny smith are taken care here. This type of analysis using predefined threshold levels also helps for major apple pulp makers like Pearl Argo in India and Bangladesh for their Appy Fizz, which are used by maceration. Even the coca-cola company invested in Indian market maceration by its Fanta apple product.

There involves 3 gateway quality analysis for the fresh export industry starting from harvest zone to the supplier (Quality analysis Level 1). Supplier to the shipper (Quality analysis Level 2) from the exporter side. And the importer(Quality analysis Level 3) during delivery. Extreme care must be taken to preserve the state of the apple from one stage to next. When the defect analysis is made by just taking the picture from different sides the damage caused to the apple is reduced. As mentioned in previous sections accuracy of workers reduces gradually at the end of the day. Usage of this tool reduces the dependency on the manual labour force.

9

Conclusion and Future Enhancement

The objective of the project has been completed by successful implementation of the proposed system. First we extracted the apple from the frame by observing green component and limiting it while image segmentation, then by observing the luminosity the defects are extracted using thresholding technique. Finally by observing processed frames the result has been concluded. We can infer that, this system can be adopted by fresh fruit/vegetable industries, farms and quality detection in import and export junctions. This system works satisfactorily for unicolor fruits and vegetables.

This system can be successfully installed in fresh fruit industries with proper hardware. Conveyor belts, rotators, light settings, camera, and power supply are main requirements for hardware implementation. Scope of project reaches only few breeds of apples. The same logic can be applied to different breeds of apples by changing few parameters. Further it can be applied for some unicolor bulk fruits and vegetables, which can be sorted by surface inspection. The amount of defected area can be calculated with respect to the area of apple approximately. By using infrared cameras, internal defects within few millimeter depth can also be detected with the above mentioned logic.

References

1. "Image analysis for apple defect detection" Czeslaw Puchalski, Jozef Gorzelany, Grzegorz Zagula, Gerald Brusewitz Department of Production Engineering, University of Rzeszow, M. Cwiklinskiej, Poland
2. Digital Image processing by Gonzalez Woobs
3. Image processing, Analysis and machine version by Sonka Hlavac Boyle
4. http://homepages.inf.ed.ac.uk/rbf/HIPR2/glossary.htm
5. http://weblog.benjaminsommer.com/blog/2012/05/19/introduction-to-image-thresholding/
6. http://www.slideshare.net/Rucku/noise-remove-in-image-processing-by-fuzzy-logic
7. www.naun.org/multimedia/NAUN/computers/20-269.pd
8. "Overview of Image Processing" by K.M.M. Rao, Deputy Director, NRSA, Hyderabad-500 037
9. http://lodev.org/cgtutor/filtering.html
10. http://en.wikipedia.org/wiki/Color_depth
11. Activity assessment of fruits using the methods of inertia moment and absolute value of the differences Md Zaheer Ansari1,a and A.K. Nirala2,b Biomedical Optics Laboratory, Department of Applied Physics, Indian School of Mines Dhanbad, Jharkhand-826004, In
12. Gabor feature-based apple quality inspection using kernel principal component analysis by Bin Zhu a, Lu Jiang a, Yaguang Luo b, Yang Tao a
13. Multispectral Method For Apple Defect Detection Using Hyperspectral Imaging System By Tao

14. http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/
15. Multispectral Image Processing and Pattern Recognition Techniques for Quality Inspection of Apple Fruits paper published by Devrim Unay
16. Using Statistical Histogram Based EM Algorithm for Apple Defect Detection by Ghobad Moradi1, Mousa Shamsi2, Mohammad H. Sedaaghi2, Setareh Moradi3
17. Detection of bruises on apple using neae-infrared hyperspectral imaging by R. Lu

Powered By LaTeX