UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **QUALITY INSPECTION SYSTEM FOR CANNED PINEAPPLE**

SESI PENGAJIAN: 2010/2011

Saya **NAM ZHEN KAI (870517-08-6265)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

[ ]	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
[ ]	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
[√]	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

92B, TAMAN SENTOSA,
36700, LANGKAP,
PERAK

Dr. KAMARUL HAWARI BIN GHAZALI
(Nama Penyelia)

Tarikh: **29 NOVEMBER 2010**

Tarikh: **29 NOVEMBER 2010**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : ____________________________________

Name : Dr. KAMARUL HAWARI BIN GHAZALI

Date : 29 NOVEMBER 2010

QUALITY INSPECTION SYSTEM FOR CANNED PINEAPPLE

NAM ZHEN KAI

A report submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering (Electrical and Electronic)

Faculty of Electrical and Electronic Engineering
University Malaysia Pahang

NOVEMBER 2010

"All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ______________________________

Author : NAM ZHEN KAI

Date : 29 NOVEMBER 2010

To my beloved mother and father

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to Dr. Kamarul Hawari bin Ghazali. He is my project supervisor. His broad knowledge, guidance, and encouragement have been essential, in completing my Quality Inspection System. He always guides and gives me many suggestions to develop and improve the efficiency of my system. His guidance and encouragement will go a long way in shaping my future both in my professional and personal development.

Moreover, I would like to thanks the FKEE, University Malaysia Pahang for providing me the necessary facilities to develop my Quality Inspection System. My appreciation also goes to other individuals, whose names are too many to mention, who have contributed directly or indirectly in the completion of this thesis.

Last but not least, I am forever grateful to my parents and siblings for their steadfast love and care. The dedication of this document barely touches my appreciation for them. I am also greatly indebted to all my course mates for being such understanding and open minded for sharing and discussing knowledge for my final year project.

ABSTRACT

The color image processing plays an important role in many of the applications nowadays. The variety image processing technique and the excellent performance is enhancing the human life to become more qualitative. This Quality Inspection System is mainly concern on the software development with MATLAB. The Quality Inspection System is helping LPNM to verify the quality of the cans of pineapple of the factory. It is more to manage quality control and health regulation at registered pineapple factory. The cans of pineapple will pick randomly to do the quality test. This system will scan and detect the yellowish of the pineapple in the cans via camera. The system will pass and certify only if the yellowish of the pineapple pass the standard of LPNM. After pass through all the testing, LPNM will certify and allow the cans of pineapple export to other countries. Through the final year project, it can be concluded that image processing technique can be implemented to detect grades for canned pineapple. The Quality Inspection System able to display the grades through a pop up window.

ABSTRAK

Pemprosesan imej berwarna memainkan peranan yang amat penting dalam pelbagai applikasi pada masa kini. Teknik pemprosesan imej yang berbagai-bagai dan pancapaiannya yang cemerlang telah mengkayakan kehidupan manusia. Sistem ini tertumpu kepada perkembangan perisian ke atas applikasi pemprosesan imej dengan Matlab. Kualiti Sistem Pemeriksaan ini akan membantu LPNM untuk mengesahkan kualiti tin nanas dari kilang. Sistem ini akan menetapkan kawalan kualiti dan peraturan-peraturan kesihatan dari kilang nanas berdaftar. Tin nanas akan dipilih secara rawak untuk melakukan ujian kualiti. Sistem ini akan mengimbas dan mengesan kekuningan nanas dalam tin melalui kamera. Sistem akan melancar dan menyertifikasi jika kekuningan nanas lulus piawai LPNM. Setelah melalui semua ujian, LPNM akan menyertifikasi dan mengizinkan eksport tin nanas ke negara lain. Melalui projek ini, boleh disimpulkan bahawa teknik pemprosesan imej dapat digunakan untuk mengesan kelas untuk tin nanas. Kualiti Sistem Pemeriksaan boleh memaparkan kelas nanas.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Quality control is a process by which entities review the quality of all factors involved in a production line. This is very essential to ensure the quality of products is fulfilling the specification and qualification. Visual inspection is one of the methods for quality control purpose. Visual inspection means inspection of structures and equipment by using either or all of human senses such as vision, smell, touch and hearing.

The quality inspection system is developed to help LPNM to verify the quality of the cans of pineapple of the factory. It is more to manage quality control and health regulation at registered pineapple factory. The cans of pineapple will pick randomly to do the quality test. This system will scan and detect the yellowish of the pineapple in the cans via camera. The system will pass and certify only if the yellowish of the pineapple pass the standard of LPNM. After pass through all the testing, LPNM will certify and allow the cans of pineapple export to other countries.

1.2 Problem Statement

The standard of grade for pineapple is uncertain. This is due to the relevant authority determine the grade manually. They grade the pineapple based on their experience. The database from LPNM for grading the pineapple is not complete due to not specific software or machine to grade the pineapple properly.

1.3 Objective

The objective of this project is to develop a system to test the quality of canned pineapple, standardize for color detection and use MATLAB for the system development.

1.4 Scopes of Project

The scopes of project are to develop a complete system to undergo color detection before grading the canned pineapple. The system has to able to analysis the yellowish of the pineapple. The whole system will undergoes real time analysis.

CHAPTER 2

THEORY AND LITERATURE REVIEWS

2.1 Introduction

This chapter includes the review of the fundamentals for various images, RGB color space and relevant mathematics. Projects implemented with color image processing and machine vision system are discussed.

2.2 Images Type

The MATLAB function toolboxes have supports four types of images, which are intensity images, binary images, indexed images and RGB images.

2.2.1 Intensity Images

Intensity image is a data matrix whose values have been scaled to represent the intensities. When the elements of an intensity image are of class uint8, it has the values in the range of [0, 255] based on the calculation of 2^8; class uint16, it has the range of [0, 65535] based on the calculation of 2^16. If the images of class double, the values are floating-point numbers. Figure 2.1 below shows the Intensity image.

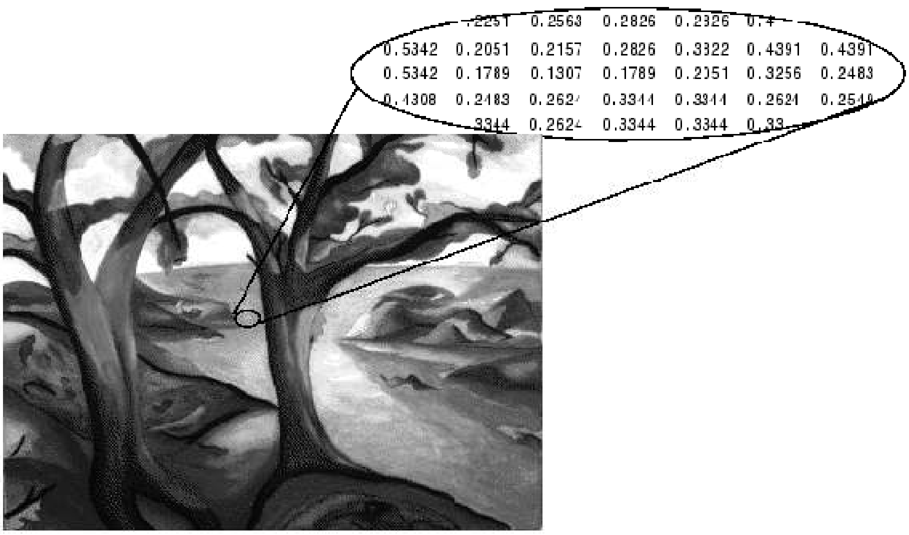


Figure 2.1 Intensity Image.

2.2.2 Binary Images

A binary number in Matlab is represented by 0s and 1s. Normally, function logical is applied to convert numerical array to binary. If A is a numeric array consisting of 0s and 1s, the logical B array will be create as B = logical (A). Figure 2.2 below shows the Binary image.

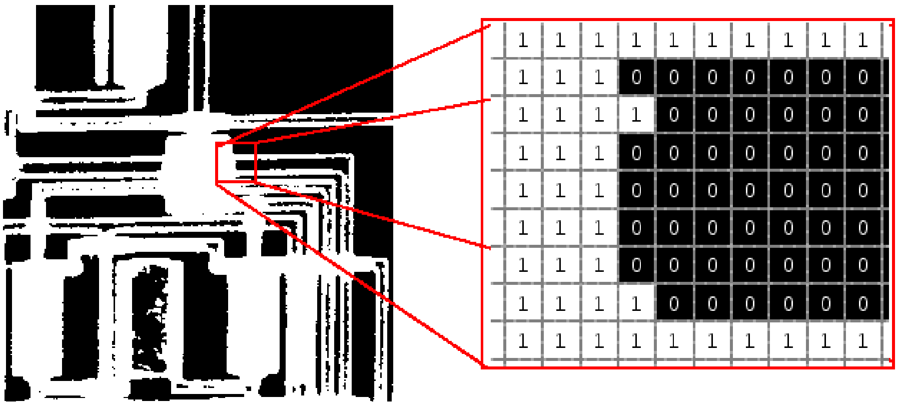


Figure 2.2 Binary Images

2.2.3 RGB Images

RGB is the 3 basic color pixels corresponding to Red, Green and Blue. A RGB color Image is an M x N x 3 array of color pixels. If an RGB image is of class double, the range of value is [0, 1]. Similarly, there are classes uint8 or uint16 for the range of [0, 255] based on the calculation of 2^8 and [0, 65535] based on the calculation of 2^16. Figure 2.3 below shows the RGB image.

Figure 2.3 RGB Image.

2.3 Basic RGB Color Space

The RGB color space is the most frequently used color space in the term of image processing. This color space is the basic one, as normally color camera, scanners and displays are most often provided with direct RGB signal input or output. The color gamut in RGB space forms a cube as shown in Figure 2.4. Each color, which is described and displayed by its RGB components, is represented by a point and can be found either on the surface or inside the cube. Because the image uses 8 bits per primary color channel, the range per primary is 0 to 255 (2^8=256).

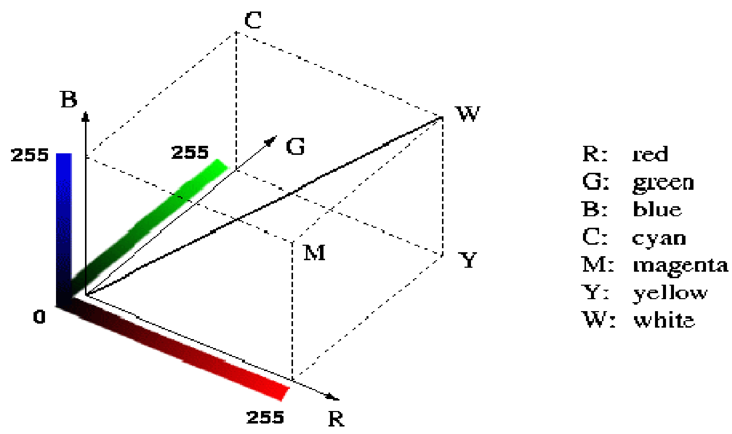


Figure 2.4 RGB color cube for a uint8 image.

The values of red, green and blue are displayed and represented at each axis of the cube in the range [0,255]. The red axis, labeled R, shows the associated color scale beneath it. The green and blue axes are illustrated similarly. The color's intensity is represented by the values between 0 and 255. Colors act like vector in the term of color space. They can be combined by addition and subtraction to obtain other colors in the cube. Thus, black color or the origin of the cube, is represented by $0_R\,0_G\,0_B$. The far corner from the origin is the sum of the highest intensities of red, green, and blue, or $255_R\,255_G\,255_B$, which is white color. All grey colors are placed on the main diagonal of this cube from black (R=G=B=0) to white (R=G=B=255).

RGB components in an image are proportional to the amount of light incident on the scene represented by the image. In order to eliminate the influence of illumination intensity, so called Chromaticity coordinates.

$$r = \frac{R}{R + G + B}$$

$$g = \frac{G}{R + G + B}$$

$$b = \frac{B}{R + G + B} = 1 - r - g$$

2.4 Applications

The measurement systems based on images can apply in various kinds of scientific areas. Besides that, in various industrial, medical and agricultural environments the applications are rapidly growing. The classification and detection in the term of color images is essential for many applications, such as image matching, image segmentation, visual tracking and object recognition in the fields of computer vision and image processing.

2.4.1 FITTING A PINEAPPLE MODEL FOR AUTOMATIC MATURITY GRADING

In this paper, we present a pineapple skin model and a method to fit the model. Our main application of the model is for automatic maturity grading of pineapples in canned pineapple industry. The model consists of two subparts: Phyllotaxis and pineapple scale models. The Phyllotaxis model represents the spiral arrangement of pineapple-scales, which is a growing pattern of the fruit. It includes a string of scale-model cells. The scale model includes boundary, internal area and petal part of scale. Modified snake algorithm is used to construct the structure model while Active Shape Model (ASM) is applied to each scale. The model can accurately fit to pineapple skins in our experiment and classification features of the fruit can be extracted.

Researches in food engineering [1] suggested that their maturity, which is closely related to their color and texture, can be identified by their physical properties and external appearance. This agrees with the way pineapple grading experts employ to classify pineapples for commercial uses. Currently the grading process in most manufacturers is done by human inspectors. They rely on visual appearance of the fruit on transmission belt. The external appearance used to determine the maturity includes shape, color, texture and relative locations of their scales [2]. Examples are color in internal area of the scale and ratio of scale sizes between lower and higher spiral levels.

2.4.2 Skin Detection using Color Pixel Classification with Application to Face Detection: A Comparative Study

This paper presents a comprehensive study of the pixel-based skin color detection techniques. Two main issues of the skin detection are the selection of the best color space and skin color pixel classification algorithm. A large set of XM2VTS face database is used to examine whether the selection of color space can enhance the compactness of the skin class and discriminability between skin and non-skin class in thirteen color spaces and six different skin color pixel classification algorithms. The results show that 1) the selection of the color space can improve the skin classification performance 2) the segmentation performance degrades only when chrominance information is used for classification 3) Bayesian classifier is found to perform better as compared to other classification algorithms. Piecewise linear decision boundary classifier algorithm outperforms all the other skin classification algorithms when it is used for images with good illumination conditions.

Detecting human skin regions in an image is the paramount importance in many computer vision areas like face detection, face recognition, and video surveillance [3], [4]. The skin detection techniques involve the classification of each image pixel into skin and non-skin categories on the basis of pixel color [5], [6]. The rationale behind this approach is that the human skin has very consistent colors which are distinct from the colors of other objects. Color is a powerful cue that can be used as a first step in skin detection because of its advantages: low computational cost, robustness against illumination changing and geometrical transformation.

When building a system that uses skin color as a feature for face detection, two issues must be addressed: what color space to select, which color skin classification algorithm to use. This paper covers both of the questions. In the past few years, a number of comparative studies of skin color pixel classification have been reported. Jones and Rehg [4] used the Bayesian classifier with the histogram technique for skin detection. Terrillon et al. [7] compared Gaussian and Gaussian mixture models across nine chrominance spaces. Greenspan et al. [8] compared skin segmentation in eight color spaces.

In this paper, a comprehensive study of two important issues of the color pixel classification approach to skin segmentation, namely selection of color spaces and classification algorithm are presented. Thirteen different color spaces and six different color pixel classification algorithms are examined for skin detection. To support this study, a large set of XM2VTS image database consisting of more than 400 color images together with manually prepared ground-truth for skin segmentation and face detection were used. The paper is organized as follows. Section 2 is devoted to different skin classifications techniques like color representations and color pixel classification algorithms. The results of our analysis and comparison are presented in Section 3, and conclusions are given in Section 4.

2.4.3 Recognition and Extraction Algorithm Design for Defect Characteristics of Armorplate Flaw Detection Image

Aiming at the detection image of strip steel which rolled on the 15mm plate production line in a steel plant, the defect characteristics recognition and extraction algorithm had been analyzed and designed, based on the computer image processing and pattern recognition theory. And the corresponding defect characteristics recognition and processing program had been programmed by vc++6.0 computer language. In the paper, the 8 direction pixel gray value search algorithm had been compiled based on the computer image color grading theory firstly, then to extract every gray level pixel information of the armor-plate detection image, and to carry out the corresponding every gray level pixel distribution probability statistic. Based on the statistical results, the two-dimension histogram Fish evaluation function algorithm for the armor-plate CCD image processing had been designed, and the result of practical application shows that the defect characteristics recognition system which programmed based on the algorithm ahead can accurately recognize and extract the defect characteristics data from the armor-plate rolled detection image, and can effectively satisfy the industrial production requirement of plate rolled.

2.4.4 Machine Grading and Blemish Detection in Apples

Five classifiers including the K-means, Fuzzy c-means, K-nearest neighbour, Multi-Layer Perception Neural Network and Probabilistic Neural Network classifiers are compared for application to color grade classification and detection of bruising of

Granny Smith apples. A number of suitable discriminate features are determined heuristically for the categorization of four classes including: high grade fruit, high grade fruit with bruising or blemishes, off-grade hit , and off-grade fruit with bruising or blemishes. Robust features based on intensity statistics are extracted from enhanced monochrome images produced by special transformation from original RGB images. The best of the five classifiers using the optimal feature set, is shown to outperform human graders viewing the same images.

This paper describes a vision system composed of a color image processing algorithm coupled with a classifier designed to perform both grading and blemish detection of Granny Smith apples. The algorithm first extracts the apple area from the image background and then a feature vector is created from the color image of the apple surface before passing it to the classifier. The design uses localized techniques in the preprocessing stage combined with a global feature extraction algorithm.

2.4.5 Parquet Sorting and Grading Based on Color and Texture Analyses

In this paper a computer vision algorithm for automatic parquet slab sorting is described, as a part of a real time automatic parquet slab sorting system. Various computer vision algorithms and methods for automatic visual inspection and automatic classification have been analyzed. Developed algorithm consists of three main stages: color analysis, texture analysis and defects detection. The color analysis is based on the percentile values obtained from the cumulative histogram of the image and texture analysis is based on the second order statistical features obtained from gray level co-occurrence matrix.

Detection of defects is implemented as the segmentation method, based on the adaptive binary threshold algorithm, which is based on a local square regions and connected component analysis methods. This way we have achieved a very accurate classifying process with about 90 percent of accuracy, which greatly outstands results of human inspector, that are about 60-70 percent.

In many manufacturing processes visual inspection plays an important role in the quality control. The use of human workers in routine tasks like this should be avoided if possible, because results in manual inspection are often worse than one could expect, since the performance of a human inspector has a strong tendency to drop radically in uninteresting jobs. A human inspector is also quite insensitive to gradually occurring small changes. Human made grading is often inconsistent.

2.4.6 High Accuracy Object Detection and Following in Color Images for Automotive Applications

This paper focuses on complex methods for object detection and following in color images. The chosen environment is the automotive environment and the targeted objects are both passenger cars and trucks. Besides the presentation of the used algorithms, the advantages of using multiple algorithms for the detection and confirmation of objects are presented. The major plus of the system is that objects up to 200 meters can be followed accurately. Therefore the detection and following of far objects is discussed in detail.

The use of color to filter out the vegetation and improve the following is presented. The methods of computing the 3D position of objects are discussed briefly. In the end some results are presented along with the future research directions. The most important requirement to an image processing system for in-car applications is to work in real-time. Due to the amount of data in color images, current solutions either prefer grayscale processing or use relatively simple algorithms. Therefore a great amount of information is often neglected. Object detection algorithms focus mainly on one or few features of an object.

This paper tries to show the benefits of a more complex approach, in which many relatively simple algorithms are combined to take advantage of the different features present in the image for a whole time in which an object remains in the camera field of view. It also shows how some general image processing techniques like image correlation can be used in real-time for the object detection.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this section methodology is divided into two parts which are for the overall system development and programming development. Methodologies are one of the most important elements to be considered to make sure that the development of the project is smooth and get the expected result. It is important to achieve the objectives that have been stated or in other words the project follows the guideline based on the objectives. Figure 3.1 show the flow chart of methodology of the overall system development and figure 3.2 shows the flow chart for the programming development.

3.2 Flowchart for Overall Project

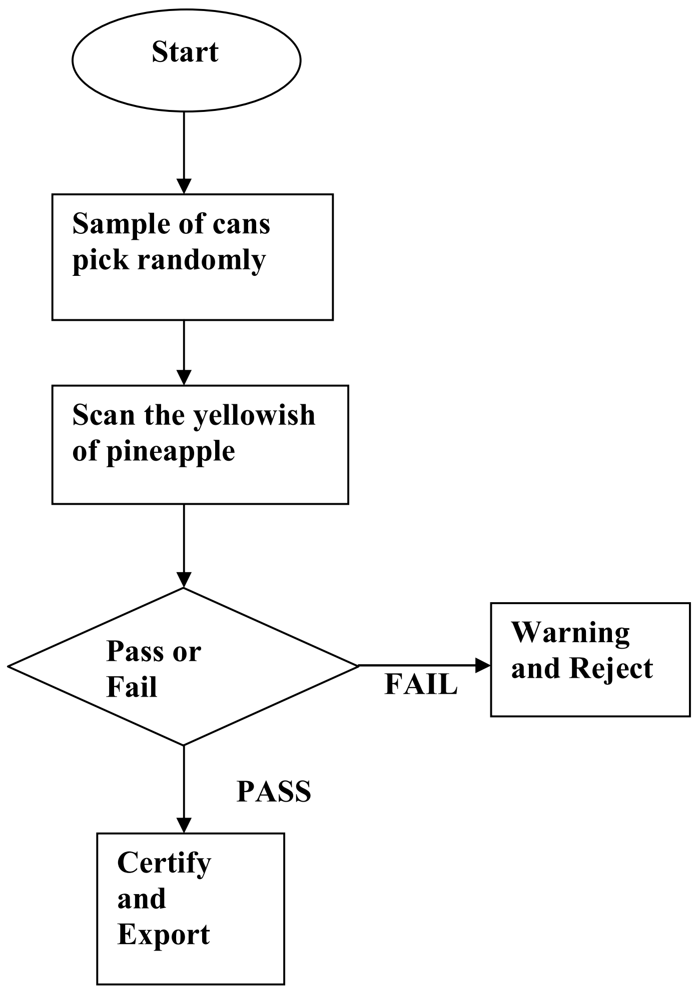


Figure 3.1: Flow Chart of the overall system

The flow chart above shows the research methodology that involved in this project. During the starting stage, the cans of pineapple will send to LPNM from several factories. The person in charge or the staffs will send all the cans to one department for quality inspection. It is to make sure that all the pineapples in cans are high quality and quality control.

After that, the cans of pineapple will pick randomly from each box. This is to make sure that the factories not mix those low grades of canned pineapples with those high qualities. It is one of the ways of quality control. Those cans have been chosen will open up and put into a container.

Then, the container will send to the conveyor of the quality inspection system. The standard of yellowish is provided by LPNM. If the pineapple passes through the test, LPNM will certify and allow the cans of pineapple to sell in the market or export. If the yellowish is below the standard of LPNM, LPNM will reject and send back all the cans to the factory which produced them. In the same time, LPNM will give warning to the relevant factory.

3.3 Flowchart for Programming Development

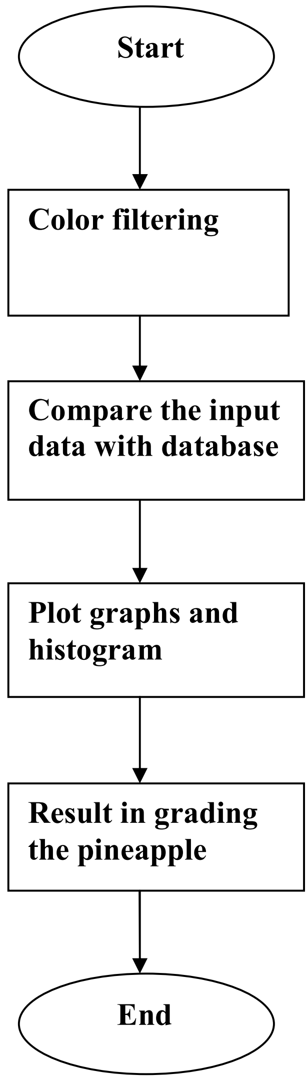


Figure 3.2: Flow Chart of programming development

Stage 1 of the quality inspiration system is filtering. Before the stage 1, the input image has to change from RGB to gray scare. The coding will use looping method to detect the yellow pixels in the images. The range of yellow pixels in the thresholding of RGB can be set manually by the user.

The coding will detect the entire single yellow pixels in the image. The color other than yellow pixels will be set to 0 and change to black. After this stage, only the pineapple in the image will remain and the background will be in black color. The stage will be useful and essential for the following stage.

One of the images will be selected as standard to do comparison at the following stages, The standard image will go through stage 1 for filtering purpose as well. It is to ensure the background pixels will be set to 0 and change to black. It is very important for grading the pineapple in high accuracy and effective.

The stage 2 will be comparing the input image with the standard image. Method of subtraction will be used in this stage. After the stage 1, both of the images have been filtered and go through the subtraction method. Figure 5 will show the image after subtraction. The stage is essential for the following stage for calculation purpose.

The coding in Stage 3 is to plot graphs and histogram. The coding will use the Figure 5 to plot 3 graphs based on RGB after subtraction method. The 3 graphs will show the value of Figure 5 in the term of RGB. The histogram will estimate of the probability distribution of a continuous variable for the Figure 5.

In stage 4, MSE will be calculated based on the Figure 5. The ratio of each grade (15, 16, 17, and 18) calculated before set the range for MSE in each grades. The ratio used as data base for the quality inspiration system. The coding will compare the MSE from Figure 5 with data base. Finally, the coding will determine the grade of the pineapple and show the grade with a pop up window.

3.4 Hardware

3.4.1 CMOS

CMOS (Complementary metal-oxide-semiconductor) is a technology for constructing integrated circuits. CMOS technology is applied in static RAM, microcontrollers, microprocessors and other digital logic circuits. CMOS technology is also applied in analog circuits such as data converters, image sensors and highly integrated transceivers for many types of communication.

CMOS is also referred to as COS-MOS (complementary-symmetry metal-oxide-semiconductor). The terms of "complementary-symmetry" refer to the fact that the typical digital design style with CMOS uses symmetrical pairs and complementary of n-type and p-type metal oxide semiconductor field effect transistors (MOSFETs) for logic functions.

The CMOS devices are high noise immunity and consume low static power. The significant power is only drawn while the transistors in the CMOS device are switching between on and off states. CMOS devices do not create the waste heat as other forms of logic, such as NMOS logic or transistor-transistor logic (TTL), which uses all n-channel devices without p-channel devices. CMOS also advance in the term of integration. It allows several functions on a single chip. This is the reason that CMOS won the race in the eighties and became the most used technology to be implemented in VLSI chips.

Each CMOS active pixel sensor cell has its own buffer amplifier, and can be addressed and read individually. A commonly used cell has four transistors and a photo-sensing element. The cell has a transfer gate separating the photo sensor from a capacitive "floating diffusion," a reset gate between the floating diffusion and power supply, a source-follower transistor to buffer the floating diffusion from readout-line capacitance, and a row-select gate to connect the cell to the readout line. All pixels on a column connect to a common sense amplifier.

CMOS image sensors are generally of a much simpler design; often just a crystal and decoupling. For this reason, they are easier to design with, generally smaller, and require less support circuitry. There are two categories of CMOS sensors, analog and digital, as defined by their manner of output. Analog sensors feed their encoded signal in a video format, such as PAL, NTSC, etc. The signal can be fed directly to standard video equipment. Digital CMOS image sensors provide digital output, typically via a 4/8 or 16 bit bus. The digital signal is direct, not requiring transference or conversion via a video capture card.

The phrase "metal-oxide-semiconductor" is a reference to the physical structure of certain field-effect transistors, having a metal gate electrode placed on top of an oxide insulator, which in turn is on top of a semiconductor material. Aluminum was once used but now the material is polysilicon. Other metal gates have made a comeback with the advent of high-k dielectric materials in the CMOS process, as announced by IBM and Intel for the 45 nanometer node and beyond. [10]

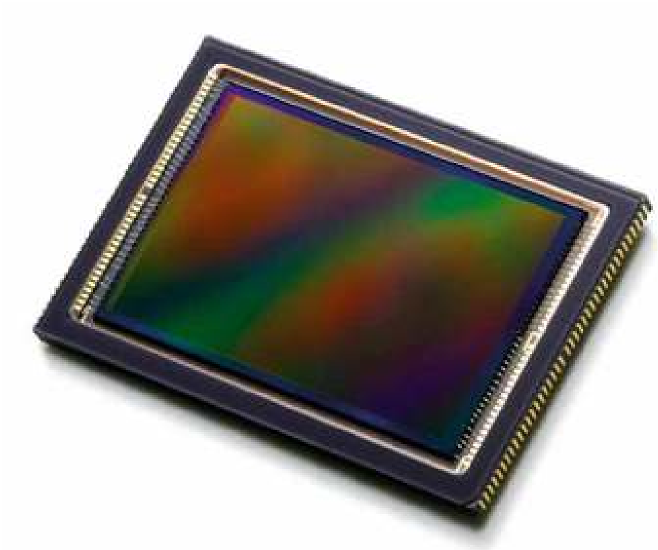


Figure 3.3: CMOS Sensor

3.4.2 CCD vs. CMOS

CCD (charge coupled device) and CMOS (complementary metal oxide semiconductor) image sensors are two different technologies for capturing images digitally. Each has unique strengths and weaknesses giving advantages in different applications. Neither is categorically superior to the other, although vendors selling only one technology have usually claimed otherwise. In the last five years much has changed with both technologies, and many projections regarding the demise or ascendency of either have been proved false. The current situation and outlook for both technologies is vibrant, but a new framework exists for considering the relative strengths and opportunities of CCD and CMOS imagers.

Both types of imagers convert light into electric charge and process it into electronic signals. In a CCD sensor, every pixel's charge is transferred through a very limited number of output nodes (often just one) to be converted to voltage, buffered, and sent off-chip as an analog signal. All of the pixel can be devoted to light capture, and the output's uniformity (a key factor in image quality) is high.

In a CMOS sensor, each pixel has its own charge-to-voltage conversion, and the sensor often also includes amplifiers, noise-correction, and digitization circuits, so that the chip outputs digital bits. These other functions increase the design complexity and reduce the area available for light capture. With each pixel doing its own conversion, uniformity is lower. But the chip can be built to require less off-chip circuitry for basic operation.

Both CCDs and CMOS imagers can offer excellent imaging performance when designed properly. CCDs have traditionally provided the performance benchmarks in the photographic, scientific, and industrial applications that demand the highest image quality (as measured in quantum efficiency and noise) at the expense of system size. CMOS imagers offer more integration (more functions on the chip), lower power dissipation (at the chip level), and the possibility of smaller system size, but they have often required tradeoffs between image quality and device cost.

Today there is no clear line dividing the types of applications each can serve. CMOS designers have devoted intense effort to achieving high image quality, while CCD designers have lowered their power requirements and pixel sizes. As a result, you can find CCDs in low-cost low-power cell phone cameras and CMOS sensors in high-performance professional and industrial cameras, directly contradicting the early stereotypes. It is worth noting that the producers succeeding with "crossovers" have almost always been established players with years of deep experience in both technologies.

Costs are similar at the chip level. Early CMOS proponents claimed CMOS imagers would be much cheaper because they could be produced on the same high-volume wafer processing lines as mainstream logic or memory chips. This has not been the case. The accommodations required for good imaging performance have required CMOS designers to iteratively develop specialized, optimized, lower-volume mixed-signal fabrication processes--very much like those used for CCDs. Proving out these processes at successively smaller lithography nodes (0.35um, 0.25um, 0.18um...) has been slow and expensive; those with a captive foundry have an advantage because they can better maintain the attention of the process engineers.

CMOS cameras may require fewer components and less power, but they still generally require companion chips to optimize image quality, increasing cost and reducing the advantage they gain from lower power consumption. CCD devices are less complex than CMOS, so they cost less to design. CCD fabrication processes also tend to be more mature and optimized; in general, it will cost less (in both design and fabrication) to yield a CCD than a CMOS imager for a specific high-performance application. However, wafer size can be a dominating influence on device cost; the larger the wafer, the more devices it can yield, and the lower the cost per device. 200mm is fairly common for third-party CMOS foundries while third-party CCD foundries tend to offer 150mm. Captive foundries use 150mm, 200mm, and 300mm production for both CCD and CMOS.

The larger issue around pricing is sustainability. Since many CMOS start-ups pursued high-volume, commodity applications from a small base of business, they priced below costs to win business. For some, the risk paid off and their volumes provided enough margins for viability. But others had to raise their prices, while still others went out of business entirely. High-risk startups can be interesting to venture capitalists, but imager customers require long-term stability and support.

While cost advantages have been difficult to realize and on-chip integration has been slow to arrive, speed is one area where CMOS imagers can demonstrate considerable strength because of the relative ease of parallel output structures. This gives them great potential in industrial applications.[11]

3.4.3 Feature and Performance Comparison

Feature	CCD	CMOS
Signal out of pixel	Electron packet	Voltage
Signal out of chip	Voltage (analog)	Bits (digital)
Signal out of camera	Bits (digital)	Bits (digital)
Fill factor	High	Moderate
Amplifier mismatch	N/A	Moderate
System Noise	Low	Moderate
System Complexity	High	Low
Sensor Complexity	Low	High
Camera components	Sensor + multiple support chips + lens	Sensor + lens possible, but additional support chips common
Relative R&D cost	Lower	Higher
Relative system cost	Depends on Application	Depends on Application
Performance	CCD	CMOS

Responsivity	Moderate	Slightly better
Dynamic Range	High	Moderate
Uniformity	High	Low to Moderate
Uniform Shuttering	Fast, common	Poor
Uniformity	High	Low to Moderate
Speed	Moderate to High	Higher
Windowing	Limited	Extensive
Antiblooming	High to none	High
Biasing and Clocking	Multiple, higher voltage	Single, low-voltage

3.4.4 CMOS Development's Winding Path

Initial Prediction for CMOS	Twist	Outcome
Equivalence to CCD in imaging performance	Required much greater process adaptation and deeper submicron lithography than initially thought	High performance available in CMOS, but with higher development cost than CCD

On-chip circuit integration	Longer development cycles, increased cost, tradeoffs with noise, flexibility during operation	Greater integration in CMOS, but companion chips still required for both CMOS and CCD
Reduced power consumption	Steady improvement in CCDs	Advantage for CMOS, but margin diminished
Reduced imaging subsystem size	Optics, companion chips and packaging are often the dominant factors in imaging subsystem size	CCDs and CMOS comparable
Economies of scale from using mainstream logic and memory foundries	Extensive process development and optimization required	CMOS imagers use legacy production lines with highly adapted processes akin to CCD fabrication

3.5 Image Processing and Software Implementation

3.5.1 Introduction

The quality inspection system is been developed via concepts of image processing. The whole program for the system is written with Matlab.

3.5.2 Image Processing

3.5.2.1 Introduction

Image processing can be explained as a physical process used to convert an image signal into a physical image. The image signal can be either analog or digital. The actual output itself can be an actual physical image or the characteristics of an image.

Photography is the most common type of image processing. An image is captured by a camera to create an analog or digital image in this process. The image is processed using the appropriate technology based on the input source type in order to produce a physical picture.

Digital image processing usually take as reference in image processing, but analog image processing and optical are possible as well.

3.5.2.2 Application of Image Processing

Image processing is being applied in the term of research and relate in such diverse fields as microscopy, astronomy, seismology, medicine, defense, the publication and entertainment industries and industrial quality control.

Image processing has expanded to include three-dimensional data sets (volume images), and even four-dimensional volume-time data sets. An example of the latter is a volume image of a beating heart, obtainable with x-ray computed tomography (CT). CT, PET, single-photon emission computed tomography (SPECT), MRI, ultrasound, SAR, atomic force microscopy, scanning tunneling microscopy, co focal microscopy and other modalities have been developed to provide digitized images directly. [12]

The digital images are widely accessible from the Internet, CD-ROMs, scanners, inexpensive charge-coupled-device (CCD) cameras, and frame grabbers. The software used for manipulating images is also widely accessible.

3.5.3 MATLAB

3.5.3.1 Introduction

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Matlab is a powerful software for multi-purpose nowadays. It is developed by the MathWorks. MATLAB allows the functions of plotting of functions and data, creation of user interfaces, interfacing with programs written in other languages, including C, C++, and Fortran, implementation of algorithms and matrix manipulations.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. The functions of adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems are additional package in MATLAB.

3.5.3.2 Interfacing with Other Languages

MATLAB is software able to call the functions subroutines and developed in the Fortran or C programming language. A wrapper function is created to allow the MATLAB data types to be returned and passed.

Libraries are written in ActiveX, NET or Java. It can be directly called from MATLAB and many MATLAB libraries are implemented as wrappers around ActiveX libraries or Java. The process of calling MATLAB from the Java is more complicated if compare to others, but it still can be done with the function of MATLAB extension, which is sold separately by MathWorks, or the users can use an undocumented mechanism called JMI (Java-to-Matlab Interface). The mechanism able to interface between Java and Matlab. [13]

As alternatives to the MuPAD based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to Mathematica or Maple.

CHAPTER 4

RESULT & DISCUSSION

4.1 Introduction

Within the two semesters of this final year project, the quality inspection system was created. This chapter will be discussed about the analysis and the corresponding results. The major focus here is to achieve an algorithm which is intelligent and robust to grade the pineapple.

4.2 Overview of the Project

The quality inspection system was designed, modified and tested within this semester. The system works perfect and grades the pineapple with high accuracy in the indoor scene with offline analysis. If the system works in real time, the percentage of accuracy will drop and takes more time for analysis.

The quality inspection system able to work either in real time or offline analysis. For the part of real time analysis, the system able to acquire image from webcam and display the image in a new pop up window. The system will read the image captured and continue for analysis purpose.

For the offline analysis, the system will pick the images manually to further the analysis. The system will analysis the images and display in histogram. The users able to read the value or figure of the images from the histogram. The following figures are the result from the offline analysis.

Figure 4.1: Input image from CMOS camera

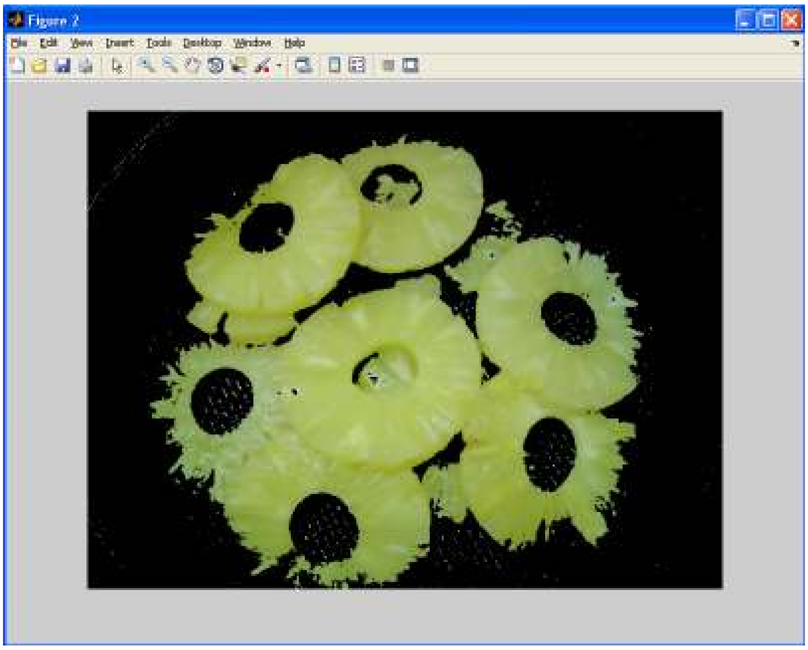


Figure 4.2: The background of the input image being filtered

Figure 4.3: Standard image for comparison purpose

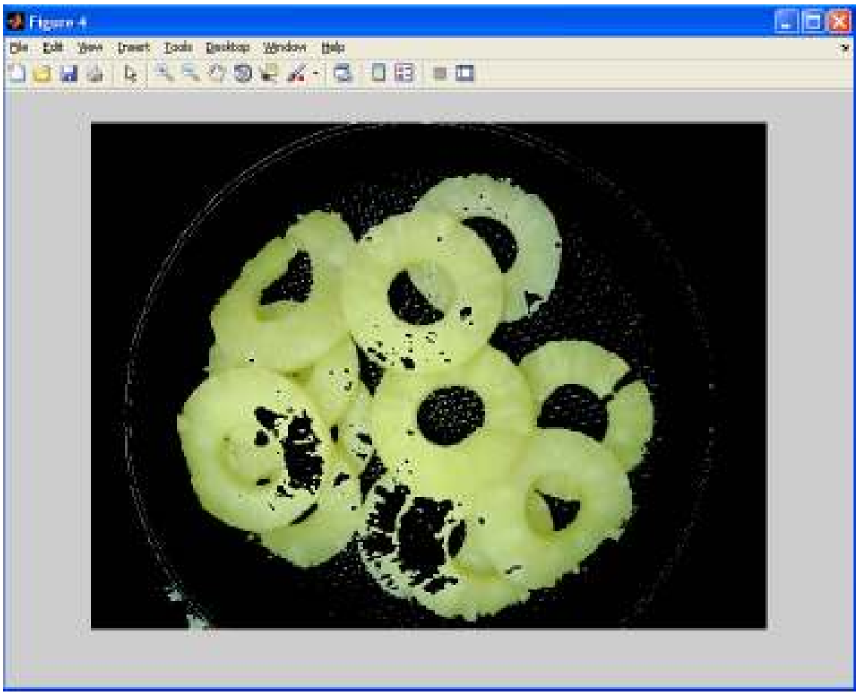


Figure 4.4: The background of the standard image being filtered

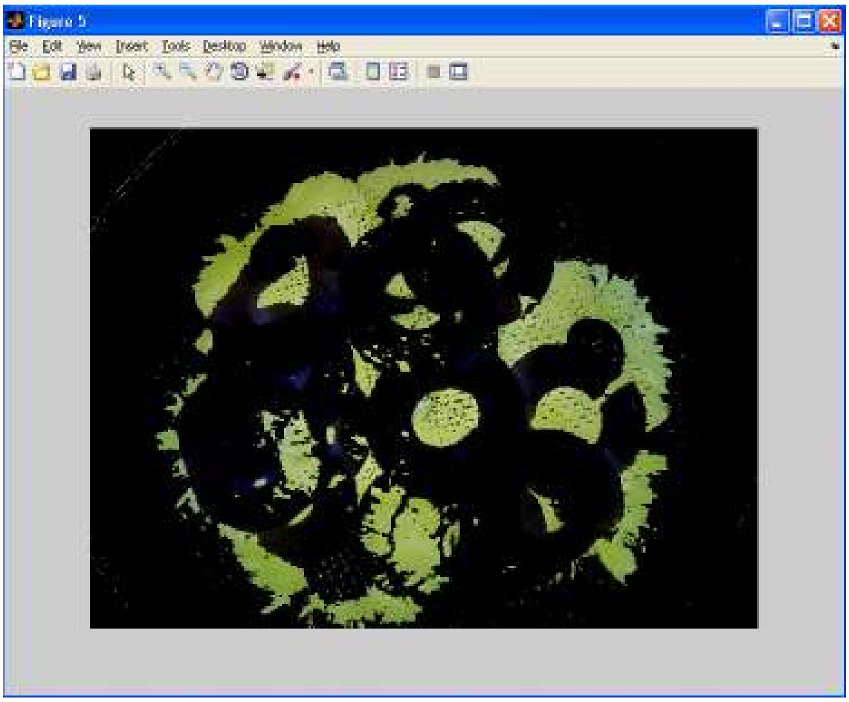


Figure 5.5: Subtraction standard image from input image

Figure 4.6: Original color image after subtraction

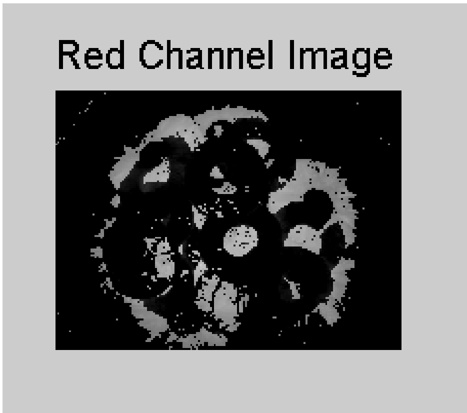


Figure 4.7: Red color image from original color image

Figure 4.8: Green color image from original color image

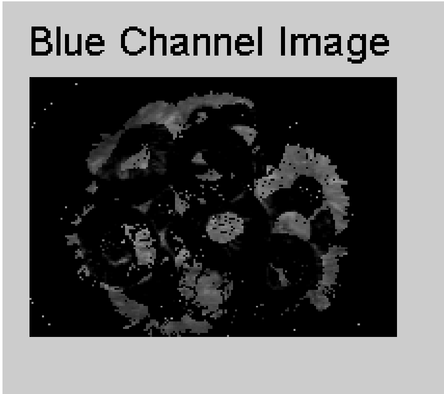


Figure 4.9: Blue color image from original color image

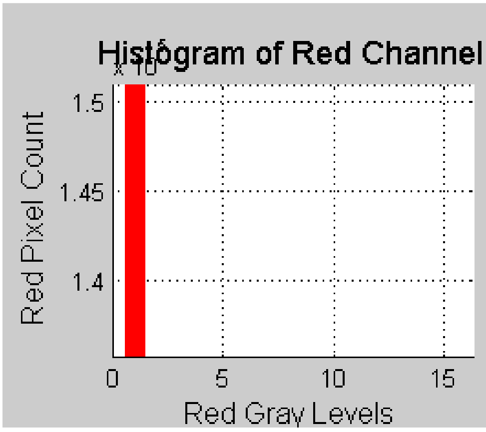


Figure 4.10: Histogram of red channel

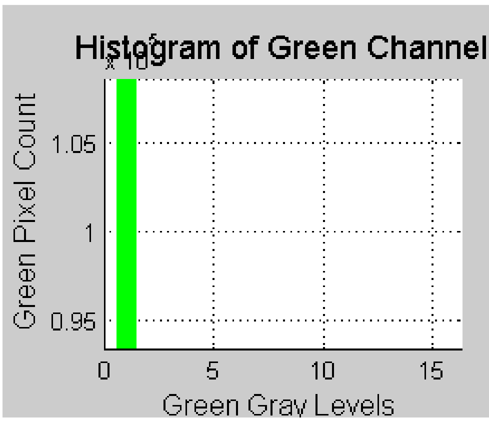


Figure 4.11: Histogram of green channel

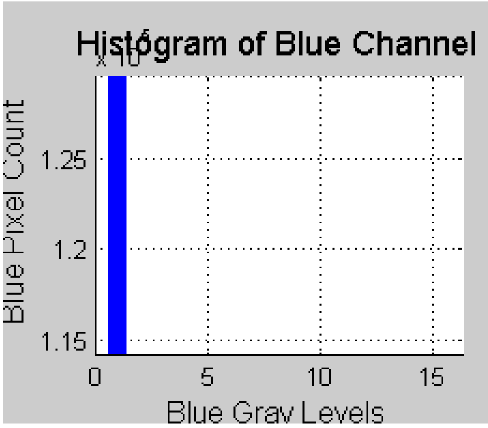


Figure 4.12: Histogram of blue channel

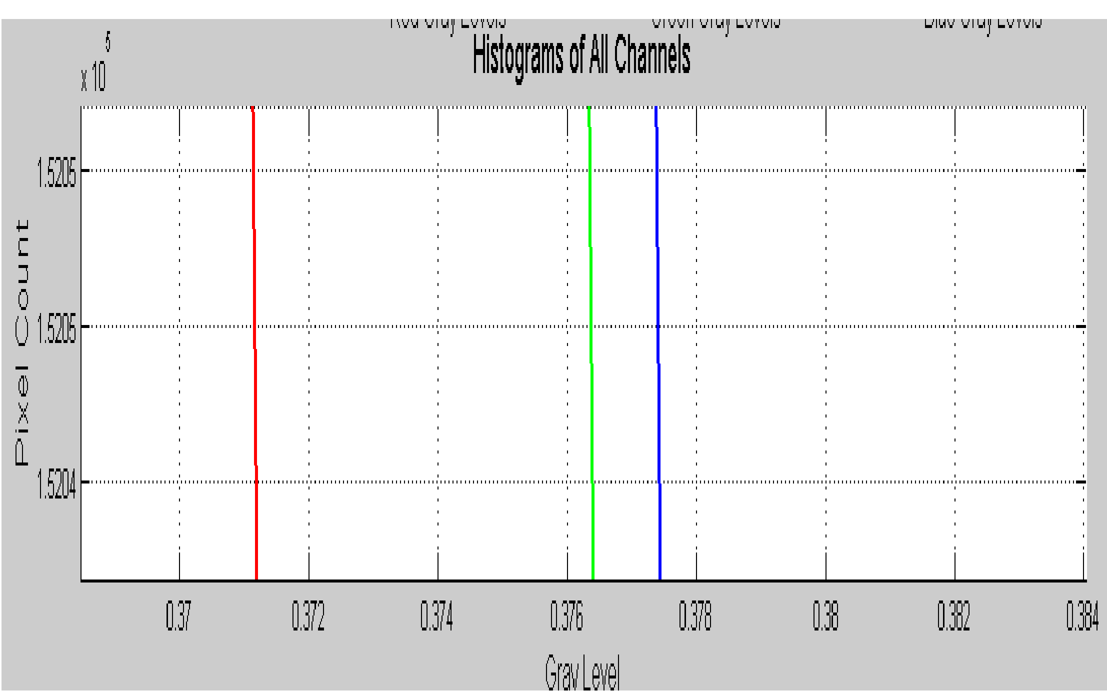


Figure 4.13: Histogram of RGB channel

Figure 4.14: Pop up window to show the grade of pineapple

4.3 Analysis

4.3.1 Try and error for thresholding the value of RGB

```
for loop1=1:r
   for loop2=1:c
      if A (loop1,loop2,1)<__A___ %A(loop1,loop2,2)
          A (loop1,loop2,1)>_B____
         A(loop1,loop2,:)=0;
      end
   end
end

for loop1=1:r
   for loop2=1:c
      if A (loop1,loop2,2)<__C___ %A(loop1,loop2,2)
         A(loop1,loop2,:)=0;
      end
```

```
    end
end

for loop1=1:r
    for loop2=1:c
        if A (loop1,loop2,3)>__D___ %A(loop1,loop2,2)
            A(loop1,loop2,:)=0;
        end
    end
end
figure(2),imshow(A)
```

The first stage of quality inspection system is filtering. I use the looping method for stage 1 due to certain reasons. The users can set the threshold based on the different situation. Threshold is the simplest method of image segmentation.

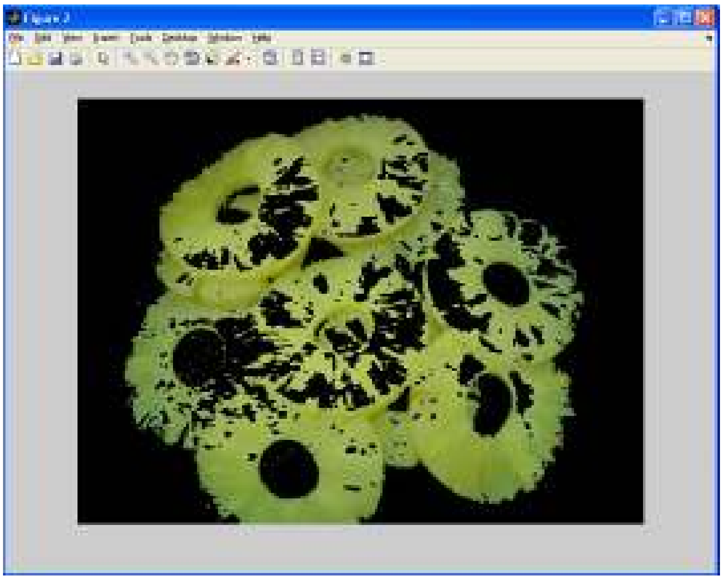
The users can set the values for A, B, C and D. As an example, the coding will detect the entire single yellow pixels in the image. The color other than yellow pixels will be set to 0 and change to black.

The users can use the method of try and error to do filtering. After the users key in the threshold values, they can observe the output from the pop up window. If they unsatisfied about the result, they can reset the threshold again and again.

The 6 sets of results below show different results due to different threshold. The users can pick the best result via observe the pop up window. The 6th set is the best threshold and best result for quality inspection system.

Set 1

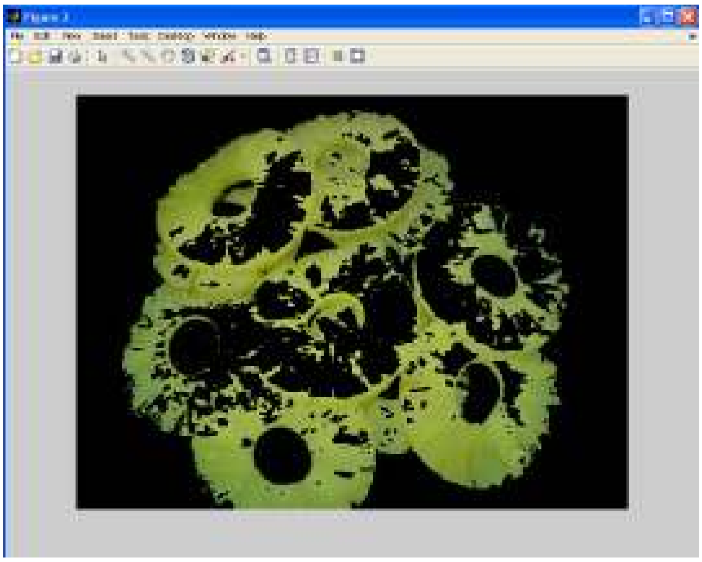
A=75 B=100 C=105 D=100



(a)

Set 2

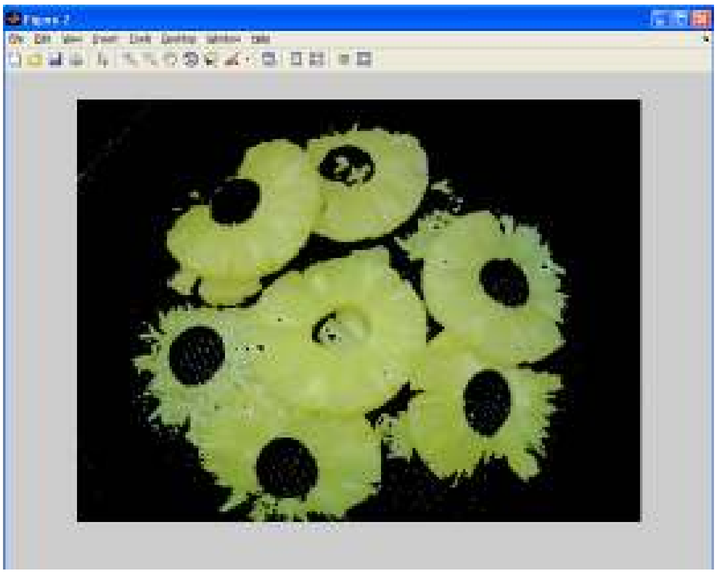
A=100 B=150 C=80 D=90



(b)

Set 3

A=70 B=190 C=150 D=150



(c)

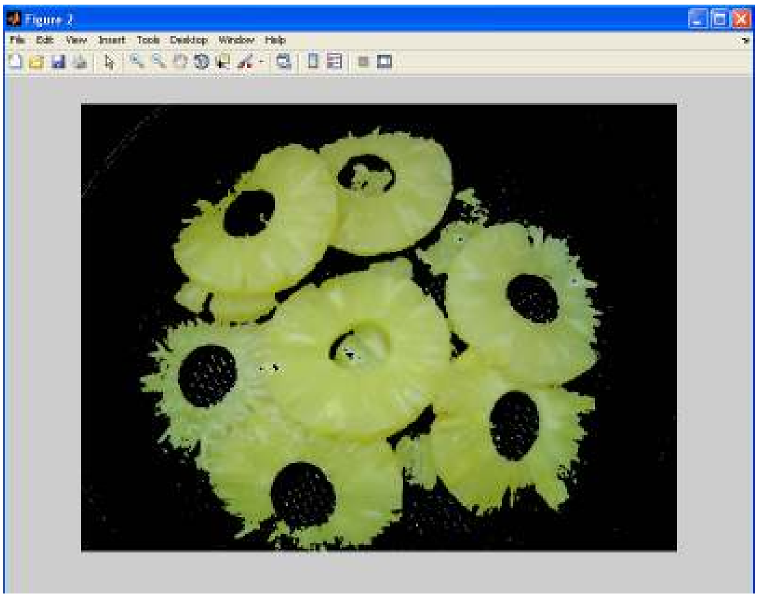
Set 4

A=100 B=180 C=140 D=160

(d)

Set 5

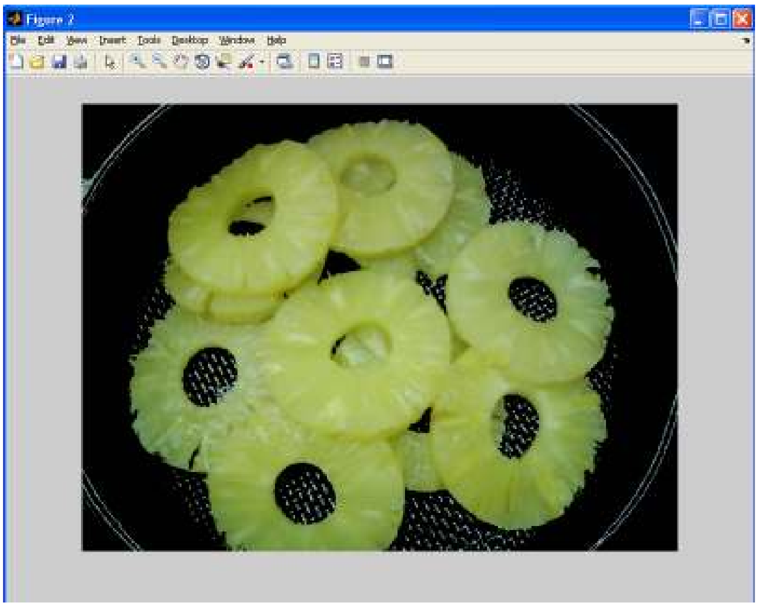
A=120 B=185 C=145 D=155



(e)

Set 6

A=90 B=160 C=100 D=180



(f)

Figure 4.15 (a-f): Testing for thresholding

```
A = double(A);
C = double(C);
D = uint8((A-C));
figure(5),imshow(D)
```

For stage 2, subtraction is one of the best solutions for comparison. This method is simple and effective. It requires less time consuming in the term of processing. The users can get the result or data from the figure 5.

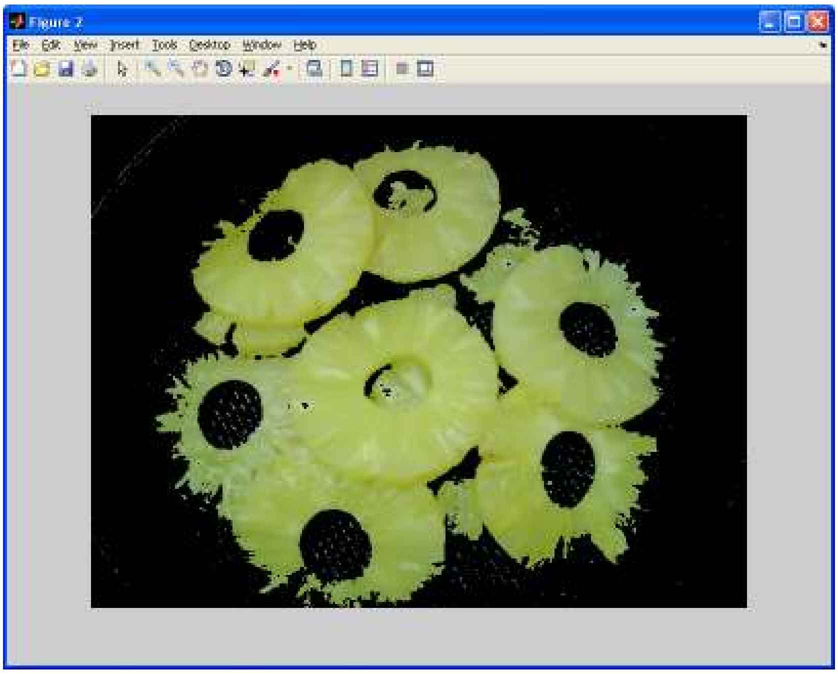


Figure 4.16: Standard image

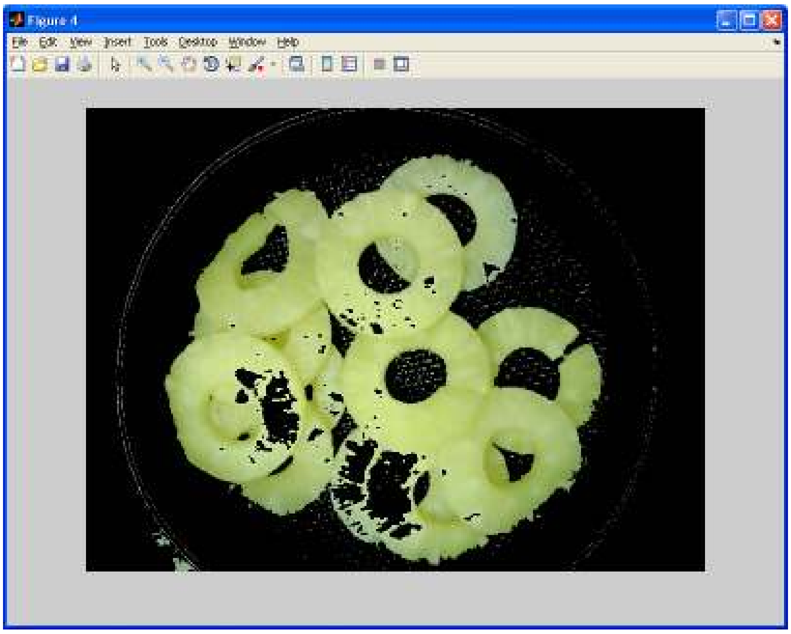


Figure 4.17 = Input image

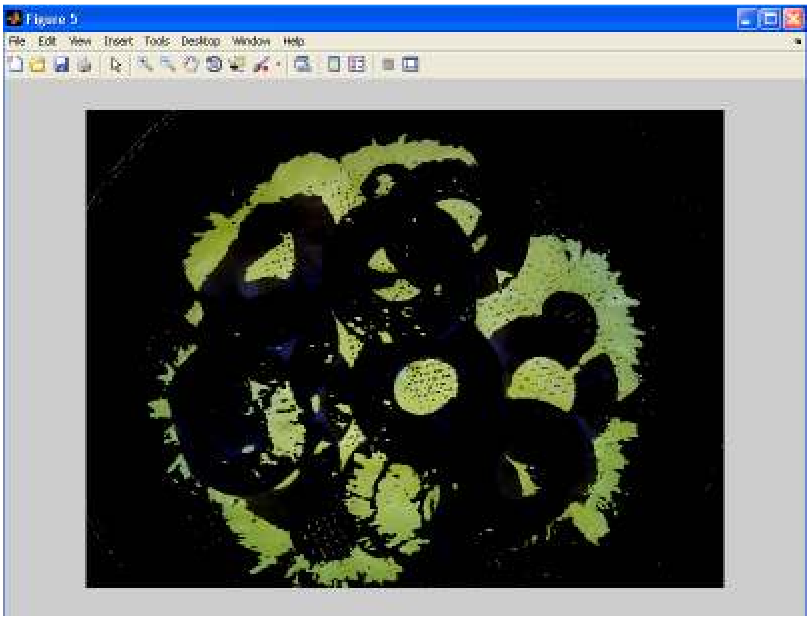


Figure 4.18: Result after subtraction

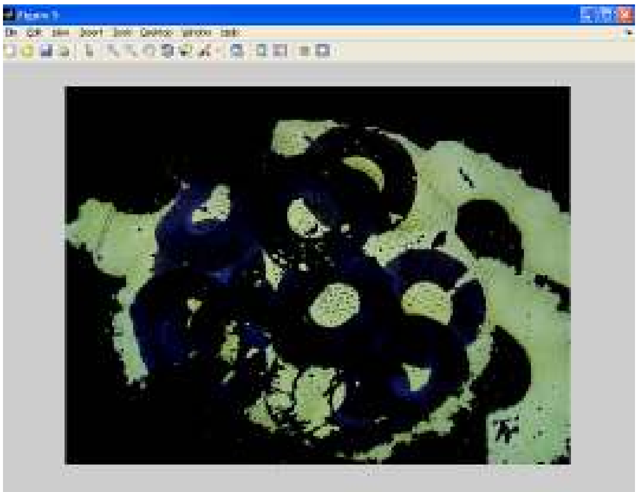
MSE calculation is used in final stage for decision making. The mean square error or MSE of an estimator is one of many ways to quantify the difference between an estimator and the true value of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the square of the "error." The error is the amount by which the estimator differs from the quantity to be estimated.

LPNM provide all the samples for grade 15, 16, 17 and 18. 30 samples will be picked from each of grades. Before select those images, quality in term of pixel values have to be taken as consideration. The images must be in high resolution. It able to ensure the output after the analysis will be in high accuracy.

After that, the MSE values of each sample have to be calculated. The values have to be recorded. It will be the data to set the range for the grading purpose. The range for grade 15, 16, 17 and 18 can be obtained from the data collected. The samples are comparing to a standard image as stated before in the stage of subtraction before the calculation of MSE. This analysis takes lot of time consuming. 30 sample images from each grade of 15, 16, 17 and 18, the total will be 120 sample images.

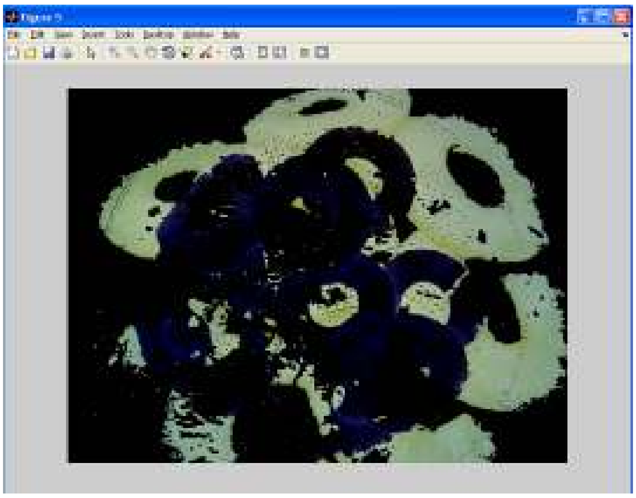
Finally, the quality inspection system is tested again after set the range of each grade. 50 samples pick randomly from grade 15, 16, 17 and 18 for grading purpose with the system. The result shows that out of 50 samples, 48 samples give the accurate result and 2 samples give failure result. The percentage of accuracy for the quality inspection system is 96 percent. Figures below show the ranges of MSE for each grades and results after MSE calculation.

Grade 15



(a)

MSE =79.0381

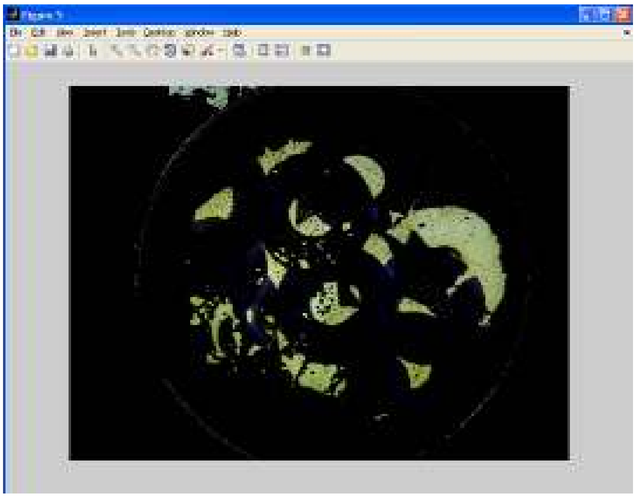


(b)

MSE = 83.4569

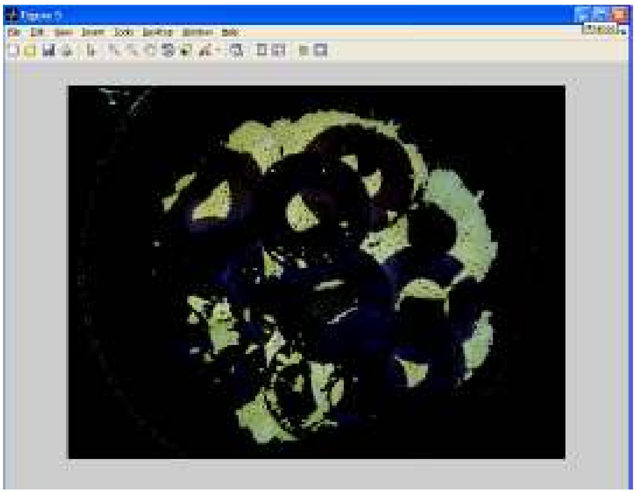
Range=79<MSE<100

Grade 16



(c)

MSE = 24.2612

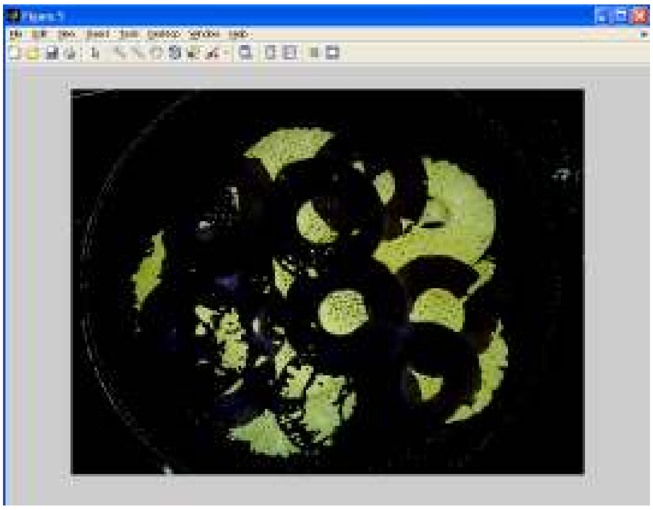


(d)

MSE = 40.2718

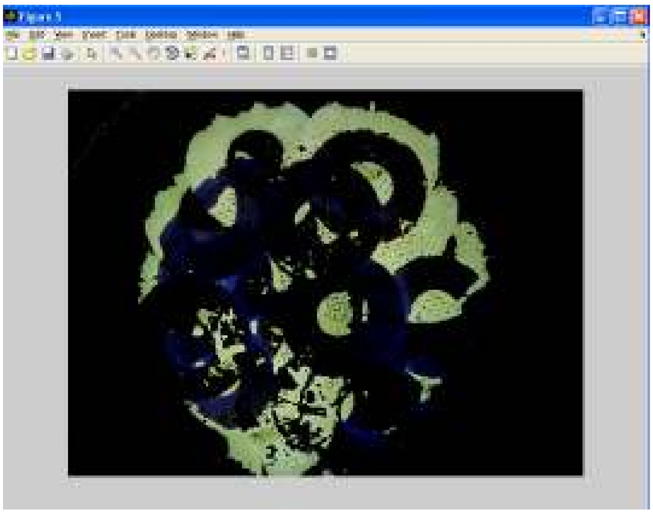
Range=24<MSE<41

Grade 17



(e)

MSE = 42.7504

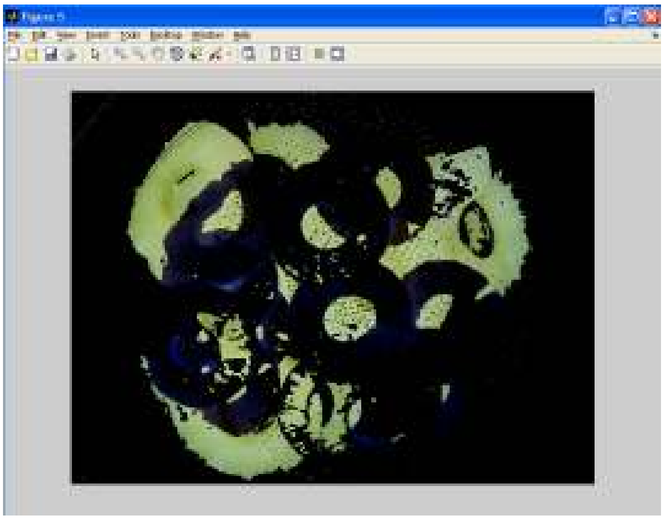


(f)

MSE = 51.7831

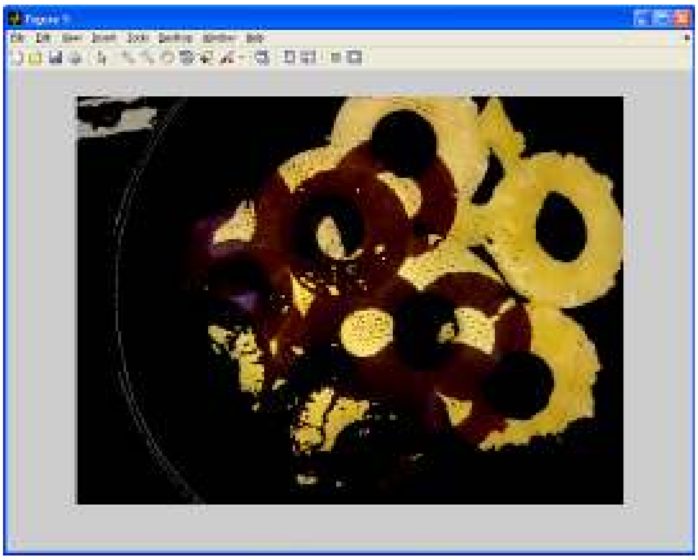
Range=42<MSE<52

Grade 18



(g)

MSE = 59.7313



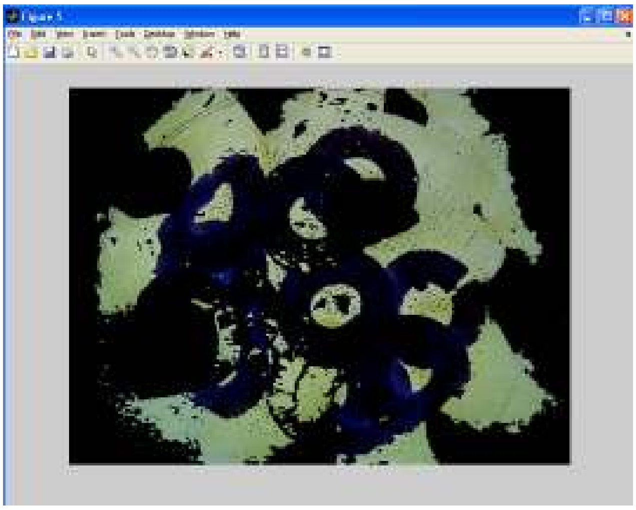
(h)

MSE =77.3248

Range=53<MSE<78

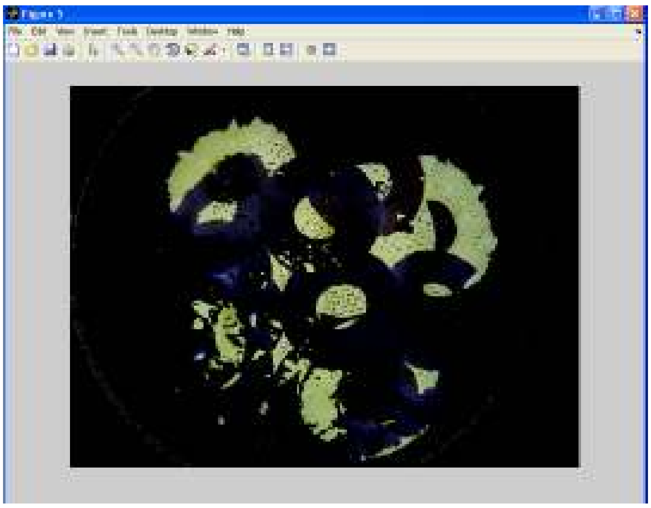
Figure 4.19(a-h): Range of MSE for each grades of pineapple

Tested Samples



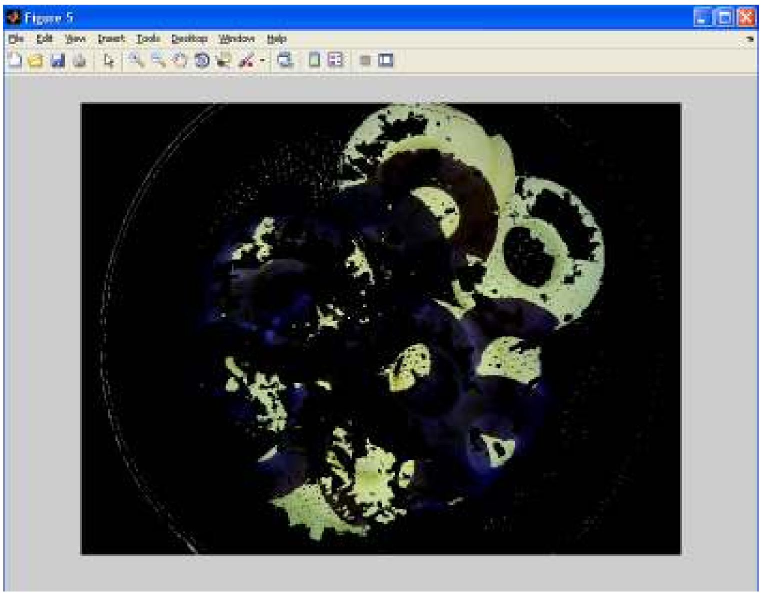
(a)

MSE = 94.2532 Grade 15



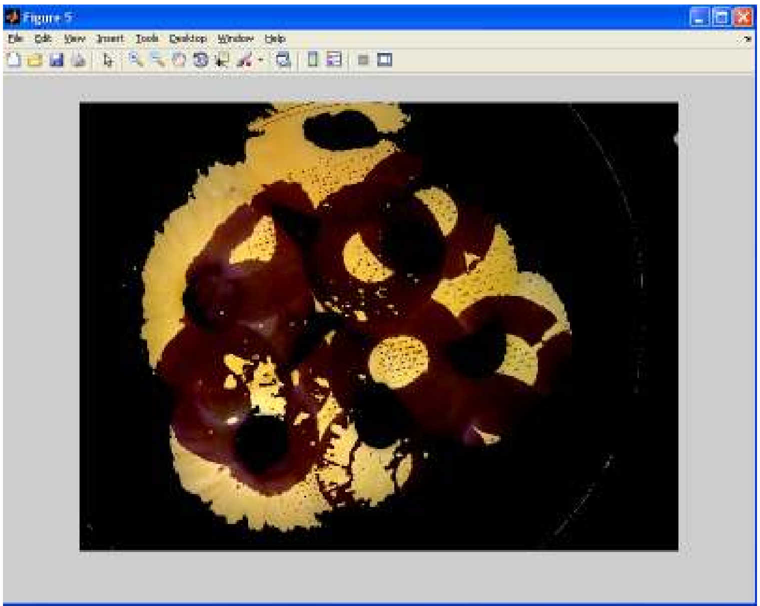
(b)

MSE =39.9876 Grade 16



(c)

MSE = 47.0101 Grade 17



(d)

MSE =67.7757 Grade 18

Figure 4.20(a-h): Result for tested samples

CHAPTER 5

CONCLUSION & RECOMMENDATIONS

5.1 Conclusion

The ideas and concepts of vision inspection are essential in daily life. As we can see, this technology has been applied in variety fields. Despite having become commonplace, the researches and developments is still keep on growing rapidly.

The quality inspection is user friendly and able to grade the pineapple in properly. The system works high efficiency and provides accurate output in a short time. Whole the process will operate in real-time. This is more advance if compare with off-line system.

In addition, a pop up window has been developed to show the grade of pineapple. The users able to read the results from the pop up window after press the run button. It takes around 30 to 40 second for the whole process. In order to increase the speed, we can use high end processor.

The system mainly provides an efficient method for quality control purpose and is aimed to be highly beneficial. As a summary of the quality inspection system, the project is done according the requirements of LPNM.

5.2 Recommendations

For future works, some recommendations have been highlighted based on the problems in order to improve the performance.

A) Improve and develop the quality inspection system into an embedded system.

B) Create 3-D for the image processing and quality inspection system. The 3-D program able to make the image more sophisticated for further analysis. The image can be move and spin in the system, so that we can capture the better figure and data from the images.

C) Develop a remote access for the quality inspection system. The users able to control the system in a certain range via remote. The result of grade will be displayed on the screen of the remote.

D) Develop a user ID and password for the relevant authority. It is to make sure the unauthorized people not able to access the system.

E) Develop a second password for the management of LPNM. The management can access the figure and adjust the threshold for the coding with the password. It is to make sure the environment issues will not affect the grading.

REFERENCES

[1] J.A. Samson, *Tropical Fruits*, 2nd ed. Longman Scientific and Technical, pp. 190-215, 1986.

[2] J. Morton, *Fruits of warm climates*, Published 1987. ISBN:0961018410.

[3] R.-L. Hsu, M. Abdel-Mottaleb, and A.K. Jain, "Face Detection in Color Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5,pp. 696-707, May 2002.

[4] J. Yang and A. Waibel, "A Real-Time Face Tracker," Proc. IEEE Workshop Applications of Computer Vision,pp. 142-147, Dec. 1996.

[5] J. Brand and J. Mason, "A Comparative Assessment of Three Approaches to Pixel-Level Human Skin Detection," Proc. IEEE Int'l Conf. Pattern Recognition, vol. 1, pp. 1056-1059, Sept. 2000.

[6] J. Jones and J.M. Rehg, "Statistical Color Models with Application to Skin Detection," Int'l J. Computer Vision, vol. 46, no. 1, pp. 81-96, Jan. 2002.

[7] J.-C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu, "Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images," Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition, pp. 54-61, Mar. 2000.

[8] H. Greenspan, J. Goldberger, and I. Eshet, “Mixture Model for Face Color Modeling and Segmentation,” Pattern Recognition Letters, vol. 22, pp. 1525-1536, Sept. 2001.

[9] http://www.dalsa.com/corp/markets/ccd_vs_cmos.aspx

[10] www.polyera.com/latest/the-benefits-of-cmos

[11] www.dalsa.com/corp/markets/ccd_vs_cmos.aspx

[12] www.answers.com/topic/image-processing

[13] www.scribd.com/doc/41484389/Eee-Latest-Technology

APPENDICES

```
clc
clear all
A=imread('D:\FORMAT\Pineapple\skala 17\DSC00885.jpg');
B=rgb2gray(A);
figure(1),imshow(A)
[r c]=size(B);
for loop1=1:r
   for loop2=1:c
      if A (loop1,loop2,1)<120 %A(loop1,loop2,2)
          A (loop1,loop2,1)>185
         A(loop1,loop2,:)=0;
      end
   end
end
for loop1=1:r
   for loop2=1:c
      if A (loop1,loop2,2)<145 %A(loop1,loop2,2)
         A(loop1,loop2,:)=0;
      end
   end
end

for loop1=1:r
   for loop2=1:c
      if A (loop1,loop2,3)>155 %A(loop1,loop2,2)
         A(loop1,loop2,:)=0;
      end
   end
end
figure(2),imshow(A)

A = double(A);

C=imread('D:\FORMAT\Pineapple\Fw__gambar_nanas_kaleng_bagi_penentuan_spek_
(skala_16)\DSC00887.jpg');
D=rgb2gray(C);
figure(3),imshow(C)
[r c]=size(D);
```

```
for loop1=1:r
   for loop2=1:c
      if C (loop1,loop2,1)<120 %A(loop1,loop2,2)
          C (loop1,loop2,1)>185
         C(loop1,loop2,:)=0;
      end
   end
end
for loop1=1:r
   for loop2=1:c
      if C (loop1,loop2,2)<145 %A(loop1,loop2,2)
         C(loop1,loop2,:)=0;
      end
   end
end

for loop1=1:r
   for loop2=1:c
      if C (loop1,loop2,3)>155 %A(loop1,loop2,2)
         C(loop1,loop2,:)=0;
      end
   end
end
figure(4),imshow(C)

C = double(C);
D = uint8((A-C));
figure(5),imshow(D)

figure(6),imshow(D)


%------------------------------------
% Read the image into an array.
rgbImage = D;
%------------------------------------


% Check to see if it is color or monochrome.
[rows columns numberOfColorChannels] = size(rgbImage);
if strcmpi(class(rgbImage), 'uint8')
% Flag for 256 gray levels.
eightBit = true;
else
eightBit = false;
end
% If it's monochrome, convert it to color.
```

```
if numberOfColorChannels == 1
rgbImage = cat(3, rgbImage, rgbImage, rgbImage);
end

% Set up sizes for the captions and text we will show on the figure.
captionFontSize = 13; % Font size for headers/captions of images.
axisFontSize = 12; % Font size for axes of the histogram plots.

% Display the original RGB image.
subplot(3,4,1);
imshow(rgbImage);
% Maximize figure.
set(gcf, 'Position', get(0, 'ScreenSize'));
% drawnow; % Make it display immediately.
if numberOfColorChannels > 1
title('Original Color Image', 'FontSize', captionFontSize);
else
title('Original Monochrome Image (converted to color)', 'FontSize', captionFontSize);
end

% Extract out the individual color channels
redChannel = rgbImage(:, :, 1);
greenChannel = rgbImage(:, :, 2);
blueChannel = rgbImage(:, :, 3);

% Display the red channel image.
subplot(3,4,2);
imshow(redChannel);
title('Red Channel Image', 'FontSize', captionFontSize);
% Display the green channel image.
subplot(3,4,3);
imshow(greenChannel);
title('Green Channel Image', 'FontSize', captionFontSize);
% Display the blue channel image.
subplot(3,4,4);
imshow(blueChannel);
title('Blue Channel Image', 'FontSize', captionFontSize);

% Calculate the histogram of the red channel.
hR = subplot(3, 4, 6);
[countsR, grayLevelsR] = imhist(redChannel);
maxGLValueR = find(countsR > 0, 1, 'last');
maxCountR = max(countsR);
% Plot the histogram of the red channel.
bar(countsR, 'r');
grid on;
xlabel('Red Gray Levels', 'FontSize', axisFontSize);
```

```
ylabel('Red Pixel Count', 'FontSize', axisFontSize);
title('Histogram of Red Channel', 'FontSize', captionFontSize);

% Calculate the histogram of the green channel.
hG = subplot(3, 4, 7);
[countsG, grayLevelsG] = imhist(greenChannel);
maxGLValueG = find(countsG > 0, 1, 'last');
maxCountG = max(countsG);
% Plot the histogram of the green channel.
bar(countsG, 'g');
grid on;
xlabel('Green Gray Levels', 'FontSize', axisFontSize);
ylabel('Green Pixel Count', 'FontSize', axisFontSize);
title('Histogram of Green Channel', 'FontSize', captionFontSize);

% Calculate the histogram of the blue channel.
hB = subplot(3, 4, 8);
[countsB, grayLevelsB] = imhist(blueChannel);
maxGLValueB = find(countsB > 0, 1, 'last');
maxCountB = max(countsB);
% Plot the histogram of the blue channel.
bar(countsB, 'b');
grid on;
xlabel('Blue Gray Levels', 'FontSize', axisFontSize);
ylabel('Blue Pixel Count', 'FontSize', axisFontSize);
title('Histogram of Blue Channel', 'FontSize', captionFontSize);

% Set all axes to be the same height.
maxCount = max([maxCountR, maxCountG, maxCountB]);
% Set all axes to be the same width.
maxGL = max([maxGLValueR, maxGLValueG, maxGLValueB]);
if eightBit
maxGL = 255;
end
% If there's a big spike at the last bin, it can be hard to see because
% of the box around the axes, so make the box a few bins wider.
% This will make it easier to see spikes in the last bin.
maxGL = maxGL + 5;
axis([hR hG hB], [0 maxGL 0 maxCount]);

% Plot all
subplot(3, 1, 3);
plot(grayLevelsR, countsR, 'r', 'LineWidth', 2);
grid on;
xlabel('Gray Level', 'FontSize', axisFontSize);
ylabel('Pixel Count', 'FontSize', axisFontSize);
hold on;
```

```
plot(grayLevelsG, countsG, 'g', 'LineWidth', 2);
plot(grayLevelsB, countsB, 'b', 'LineWidth', 2);
title('Histograms of All Channels', 'FontSize', captionFontSize);
maxGrayLevel = max([maxGLValueR, maxGLValueG, maxGLValueB]);
% Trim x-axis to just the max gray level on the bright end.
if eightBit
xlim([0 255]);
else
xlim([0 maxGrayLevel]);
end

dx = D; %D = uint8((A-C))
MSE = mean( dx(:).^2 ) %Mean-Square-Error

  if (MSE>79 && MSE<100)
     Grade=15;
      promptMessage = sprintf('Grade 15');
       disp(promptMessage); % Write to command window.
        uiwait(msgbox(promptMessage)); % Also pop up a message box.
         close();
  else if(MSE>24 && MSE<41)
   Grade=16;
    promptMessage = sprintf('Grade 16');
     disp(promptMessage); % Write to command window.
      uiwait(msgbox(promptMessage)); % Also pop up a message box.

     else if(MSE>42 && MSE<52)
   Grade=17;
    promptMessage = sprintf('Grade 17');
     disp(promptMessage); % Write to command window.
      uiwait(msgbox(promptMessage)); % Also pop up a message box.

        else if (MSE>53 && MSE<78)
   Grade=18;
    promptMessage = sprintf('Grade 18');
     disp(promptMessage); % Write to command window.
      uiwait(msgbox(promptMessage)); % Also pop up a message box.
        close();
  else
       Grade= 0;
         promptMessage = sprintf('Out of Range');
           disp(promptMessage); % Write to command window.
             uiwait(msgbox(promptMessage)); % Also pop up a message box.
end
end
end
  end
```