UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS[♦]

JUDUL: **PALM OIL CLASSIFICATION USING RGB AND FUZZY**

SESI PENGAJIAN: 2010/2011

Saya **KHAIRUN NOR AIMI BT GHAZALLI (860501-56-5668)**
(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.
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:	Nama Penyelia:
NO 28 JALAN SEROJA 15, **BANDAR BARU SALAK** **TINGGI, 43900 SEPANG,** **SELANGOR.**	**ROHANA BT ABDUL KARIM**
Tarikh: **26 NOVEMBER 2010**	Tarikh: : **26 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).

PALM OIL CLASSIFICATION USING RGB AND FUZZY

KHAIRUN NOR AIMI BT GHAZALLI

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti 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 : KHAIRUN NOR AIMI BT GHAZALLI

Date : 26 NOVEMBER 2010

DEDICATION

Special dedicated to

To my beloved family, lecturer, friends and those people who have guided and inspired me throughout my journey of education

ACKNOWLEDGEMENT

Alhamdulillah, His Willingness has made it possible for me to complete the final year project (Projek Sarjana Muda) in time.

I would like to take this opportunity to express gratitude to my dedicated supervisor, Mrs Rohana binti Abdul Karim for her guide that help this project at every stage and getting things done by sharing his valuable ideas and knowledge. Also very huge regard and very thankful because of credibility and commitment in give tremendous amount of advice while ongoing project Palm Oil Classification Using RGB and Fuzzy.

I would also like to thank to Dr Kamarul Hawari bin Ghazali for give me an idea and their data and also to all lecturers whom had helped directly or indirectly thus making this project a reality.

Not forgotten to my colleagues for their openhandedly and compassionately guided, assisted, and supported me to make this project successful. My deepest thanks to my dearest family which always support and pray on me throughout this project. Their blessing gave me the high-spirit and strength to face any problem that had occurred and to overcome them appropriately.

The great cooperation, kindheartedness and readiness to share worth experiences that have been shown by them will be always appreciated and treasured by me, thank you.

ABSTRACT

This research is about to classification palm oil by using RGB and Fuzzy. As we know the current practice in the oil palm is to grade the oil palm bunches manually using human graders for separate which one for producing oil or others effectiveness. This method is subjective and subject to disputes. In this case, we developed systems by using image processing technique RGB as a preprocessing and fuzzy logic as classifier. The RGB color technique is utilized as the extracted features for the oil palm fruit rind. Further, the extracted feature is classified using fuzzy logic system to determine the ripeness level of the oil palm fruit. The system has been design to act like human eye and brain by process the fruit images and made a decision based on selected category. The result shows that it was successful discriminate the fruit bunches with accuracy False Rejection Rate (FRR) about 10% and False Acceptance Rate 0% for ripe categories, 20% and 0% False Rejection Rate (FRR) and 0% for False Acceptance Rate (FAR) for under ripe and unripe categories.

ABSTRAK

Tujuan kajian ini adalah untuk mendapatkan klasifikasi buah kelapa sawit dengan menggunakan teknik RGB dan Fuzzy. Seperti yang kita ketahui pada hari ini, kaedah penggredan buah kelapa sawit adalah secara manual iaitu melalui kepakaran manusia untuk menentukan samada buah kelapa sawit itu masak,separuh masak ataupun masih muda lagi. Secara praktikalnya kaedah ini adalah sangat subjektif dan pelbagai pandangan dapat dibuat melalui mereka yang mahir. Oleh yang demikian satu kajian di laksanakan bagi meningkatkan mutu dan tahap penggredan buah kelapa sawit ini. Dengan menggunakan teknik “Image processing (RGB)” sebagai pra pemprosesan dan logik fuzzy sebagai pengklasifikasinya Teknik warna RGB digunakan untuk menentukan ciri-ciri dihasilkan untuk kulit buah kelapa sawit dan cirri yang dihasilkan dikelaskan menggunakan sistem logik fuzzy untuk menentukan tahap kematangan buah kelapa sawit. Sistem ini telah dibuat untuk bertindak seperti otak dan mata manusia dengan memproses gambar buah kelapa sawit dan membuat keputusan berdasarkan kategori yang dipilih. Oleh yang demikin, keputusannya menunjukkan bahawa system ini berjaya membezakan tandan buah kelapa sawit itu samada masak,separuh masak ataupun masih muda dengan peratusan ketepatannya berdasarkan nisbah bilangan data yang betul tapi tidak diterima oleh sistem sebanyak 10% dan data yang salah tapi sistem terima 0% bagi kategori masak, sebanyak 20% dan 0% data yang betul tapi sistem tidak terima dan 0% bagi data yang salah tetapi sistem terima untuk kategori separuh masak dan tidak masak buah kelapa sawit.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURE

LIST OF FIGURE

LIST OF FIGURE

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview Of Palm Oil

Malaysia is currently one of the world's largest producer and exporter of palm oil. Oil Palm cultivation began in 1917 but growth was initially very slow. Palm oil sector was only during the last 50 years that plantation development was accelerated through large-scale investment in the cultivation of the oil palm as one of the approved crops for diversifying the country's agricultural development.[1]. Oil palm were introduced to Java by the Dutch in 1848 and Malaysia by the British colony of Malaya in 1910 by Scotsman William Sime and English banker Henry Darby[3]. Oil palm can be categorized into two species there are Arecacea or Palma family which are using for agriculture for producing palm oil.Elaeis guineensis is oil palm from the west African countries and widely use as a cooking oil.

Oil palm industry is a vast and vital industry in Malaysia that our own country had been the largest production of palm oil export. The total oil palm planted area in the country increased by 4.5% to 4.69 million hectares in 2009. Among the regions, Sarawak registered the largest increase in planted area with a growth of 12.8%, followed by Peninsular Malaysia 3.3% and Sabah 2.1%. Sabah is still the largest oil palm planted State, accounting for 1.36 million hectares or 29% of the total planted area in the country. Exports of palm oil gained 2.9 % to 15.87 million tonnes in 2009 as against 15.41 million tonnes in 2008[4].

Nowadays, image processing is widely used in automation and manufacturing production[2][7][8][9]. The advancement in computer technology allows researches to implement it in agriculture industry. Over the last 50 years, research and development activities and technological advances have helped for improved oil palm sector. Grading of oil palm fruit bunches are developed using image processing technique. Image processing using RGB and fuzzy logic has been used as a tool to analyze oil palm fruits bunch. RGB technique is like analyze the colour of the fruit and obtain its quality based on the density of the RGB model.

Currently, grading the oil palm are using by human grader, where the grader will follow the standard inspection given by authorities based on oil palm maturity if ripe, unripe and under ripe. But human inspection are very subjective as depends on different human graders classify differently and may expert grader may fail to articulate the grading criteria properly. Besides that, for grading oil palm generally dependent upon colour of rind. The colour and lighting when taking a picture for grading will effect the classifying the maturity of oil palm bunch.

So, by developing the system that able to recognize the colour and can make decision like human thinking, it can make grading system would be easier in agriculture industry. The human eyes perceive colours differently and this very often lead to dispute between graders and sellers. Therefore, a standard colour or features of oil palm for grading will determine and it easier to make classifying of maturity oil palm bunch.

Figure 1.1 : Oil Palm tree

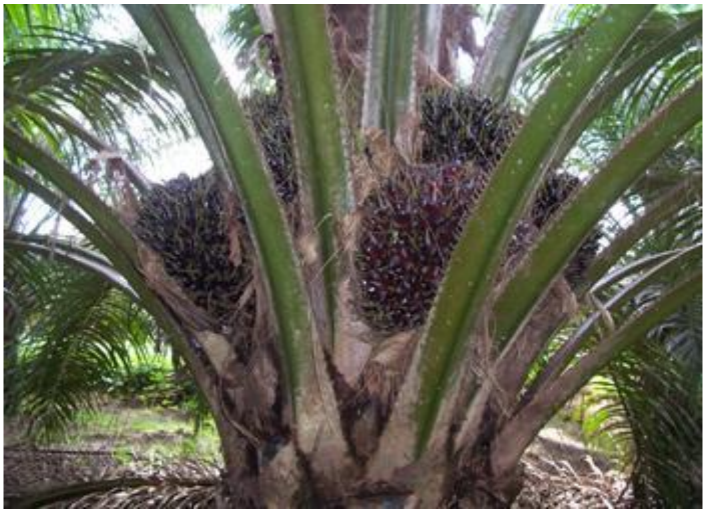


Figure 1.2 : Oil Palm bunch

1.2 Problem Statement

Agriculture industry are widely industry in Malaysia. One of the them is oil palm industry. As we know, Malaysia is one of the largest export oil palm in the world. As an export commodity, the production of good quality oil palm is vital because most of the importer countries are a quality conscious customer. Thus, an effort towards the best quality production of the oil palm should be discovered. Besides, it will complement Malaysia ambition in expanding the agriculture products to support growth in the economy. The implementation of the fuzzy logic in grading oil palm bunch can make the work grading more faster and not take a long time. So we need the good systematic system grading of oil palm for maintain and increase the producing of oil palm.

1.3 Objectives

The objective of this project are:

i. To determine maturity of oil palm fruit bunch
ii. To analyze the best structure of fuzzy logic for grading oil palm
iii. To developed algorithm for maturity grading

1.4 Scope Of Project

The focuses of this project are :

i. Classification of maturity oil palm
   - Ripe
   - Unripe
   - Under ripe

ii. Using parameter based on reddish and full bunch.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Colour provides valuable information in estimating the maturity and examining the freshness of fruits and vegetables. Colour also is one of the most significant criteria related to fruit identification and fruit quality and is a good indicator for ripeness. In agricultural applications, especially for fruits, we can estimate fruit quality from their colour. Nowadays, this technology are widely used in agriculture application and many researcher has done their research about this image processing technique. Quality features such as the color, size and defects of apples were measured through different equipment. The same set of apples was graded by both a human expert and a FL system designed for this purpose[7]. Monitor watermelon ripeness based on image processing technique and fuzzy logic as classifier. The RGB color technique is utilized as the extracted features for the watermelon's rind[8]. Oil palm Fruit Bunch Grading System Using Red, Green, Blue Digital Number was developed to distinguish between three different categories of oil palm fruit bunches. The maturity of colour ripeness index was based on different colour intensity.[9]

2.2 Manual Grading Of Oil Palm

Normally, in the agricultural industry grading method performed manually by human expertise. Therefore, the oil palm industry of Malaysia Palm Oil Board (MPOB) has issued a manual grading of oil palm fruits to promote quality awareness, especially among farm operators and small farmers. This manual are describe about the including sampling, and grading methods

Grading manual are available as a practical guide for grading of oil palm fruits and the main objectives are:

i. To improve the oil palm fruits which are marketed to the factory
ii. To improve the quality of crude palm oil production.
iii. To improve the efficiency of the extraction rate of palm kernel oil at the factory.
iv. To determine the supplier and manufacturer returns commensurate

Bunches Classification:

i. Ripe bunches had red and orange colour and outer mesocarp is orange . This bunch had at least ten (10) relay sockets fresh fruit and more than fifty percent (50%) are still attached to the bunches during the inspection at the factory. Fruit bunches and a detached fruit should be sent to the factory within 24 hours after harvest.

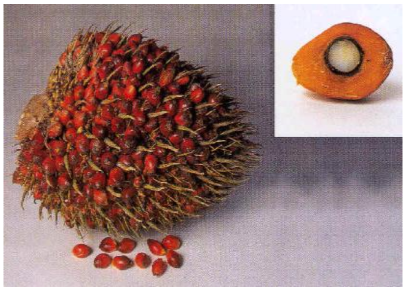


Figure 2.1: Ripe Bunch

ii. Under ripe bunches has reddish orange or reddish purple colour and the outer layer of the fruit mesocarp slightly orange color. This bunch has less than ten (10) detached sockets fresh fruit during inspection at the factory. Fruit bunches and the detached fruit should be sent to the factory within 24 hours after harvest.

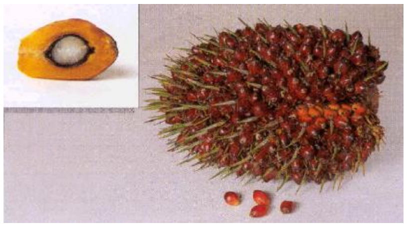


Figure 2.2 : Under ripe bunch

iii. Unripe bunches had black or dark purple and their mesocarp are yellowish at outer layer. This bunch does not had fresh fruit relay socket during inspection at the factory. Fruit relay socket (if any) in the young fruit bunches are not caused by the normal ripening process.

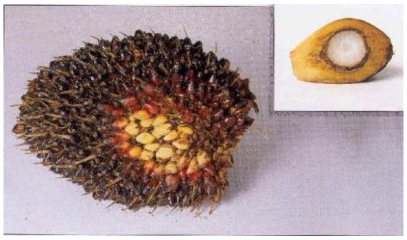


Figure 2.3 : Unripe Bunch

iv. Overripe bunches had a dark red colour and more than fifty percent (50%) had detached fruit, but at least ten percent (10%) of fresh fruit is still attached to the bunch during the inspection at the factory. Overripe bunches and the detached fruit should be sent to the factory within 24 hours after harvest.

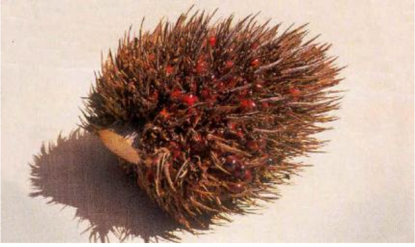


Figure 2.4 : Over ripe bunch

v. Empty bunch are more than ninety percent (90%) of fruit are detached from the bunch during the inspection at the factory.

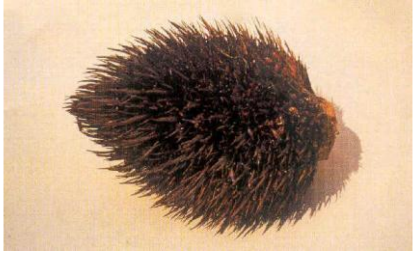


Figure 2.5 : Empty Bunch

vi. Detached fruit is reddish orange fruit that fell from the ripe bunches. It should be sent to the factory within 24 hours after the bunches are harvested.

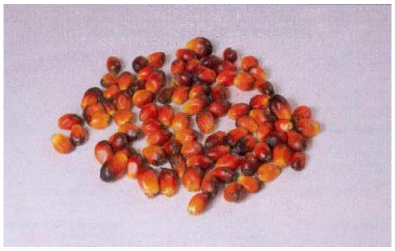


Figure 2.6 : Detached Fruit

2.3 RGB Colour Model

The RGB colour model is composed of the primary colours Red, Green and Blue, and RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors.

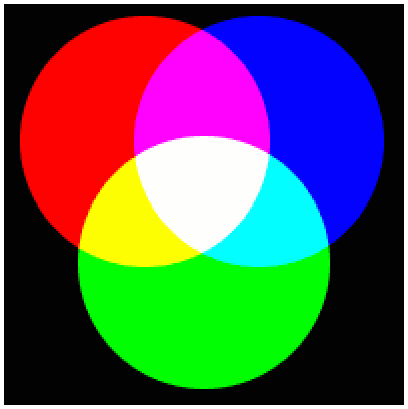


Figure 2.7 : Additive primary colour

RGB values encoded in 24 bits per pixel (bpp) are specified using three 8-bit unsigned integers (0 through 255) representing the intensities of red, green, and blue. This representation is the current mainstream standard representation for the socalled true color and common color interchange in image file formats such as JPEG or TIFF. It allows more than 16 million different combinations, hence the term millions of colors some systems use for this mode, many of them indistinguishable to the human eye [5].

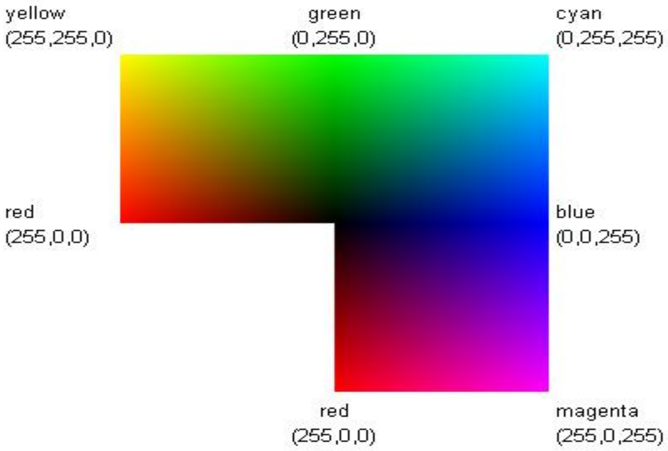


Figure 2.8 : Fully saturated faces of a 24-bpp RGB cube

2.4 Pixel

Pixel is a computer word formed from picture element, because a pixel is the smallest element of the digital image. A digital color image pixel is a RGB data value (Red, Green, Blue). Each pixel's color sample has three numerical RGB components (Red, Green, Blue) to represent the color. These three RGB components are three 8-bit numbers for each pixel. Three 8-bit bytes (one byte for each of RGB) is called 24 bit color. Each 8 bit RGB component can have 256 possible values, ranging from 0 to 255. For example, three values like (250, 165, 0), meaning (Red=250, Green=165, Blue=0) to denote one Orange pixel.

The pixel's RGB data value shows how much Red, and Green, and Blue, and the three colors and intensity levels will be combined at that image pixel, at that pixel location. The composite of the three RGB values creates the final color for that one pixel area. In the RGB system, we know Red and Green make Yellow. So, (255, 255, 0) means Red and Green, each fully saturated (255 is as bright as 8 bits can be), with no Blue (zero), with the resulting color being Yellow.

Black is a RGB value of (0, 0, 0) and White is (255, 255, 255). Gray has the property of having equal RGB values. So (220, 220, 220) is a light gray (near white), and (40,40,40) is a dark gray (near black). Gray has no unbalanced color cast. Since gray has equal values in RGB, Black & White grayscale images only use one byte of 8 bit data per pixel instead of three. The byte still holds values 0 to 255, to represent 256 shades of gray.

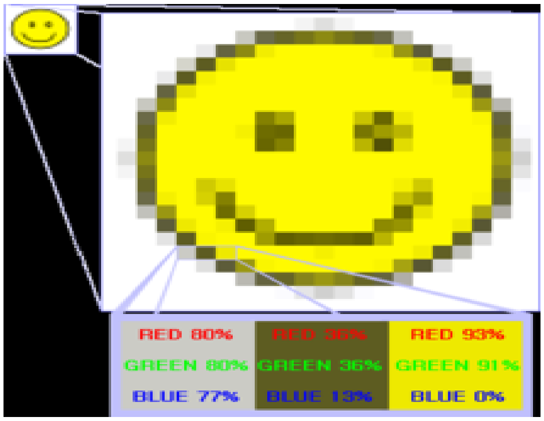


Figure 2.9 : Pixel

2.5 Colour Space

A colour space is defined as a model for representing colour in terms of intensity values. RGB color space is any additive color space based on the RGB color model. A particular RGB color space is defined by the three chromaticities of the red, green, and blue additive primaries, and can produce any chromaticity that is the triangle defined by those primary colors. Most image formats such as JPEG, BMP, GIF use the RGB colour space to store information.

2.6 Fuzzy Logic

The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology but as a way of processing data by allowing partial set membership rather than crisp set membership or non membership. Fuzzy Logic is used to handle uncertainty, ambiguity and vagueness. It provides a means of translating qualitative and imprecise information into quantitative linguistic terms. [10]

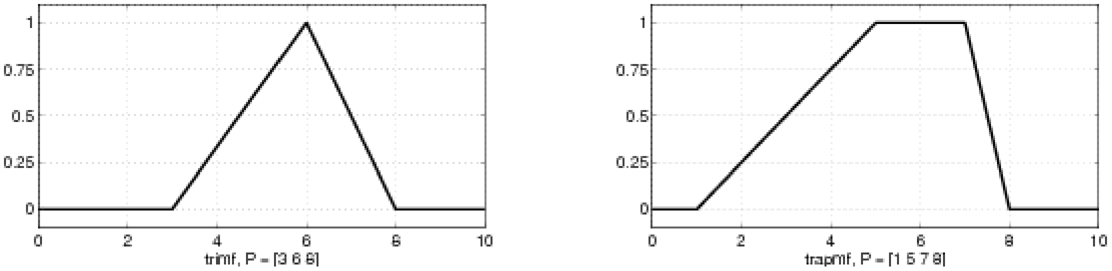
Fuzzy logic starts with the concept of a fuzzy set. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. Unlike classical set theory that classifies the elements into a continuous set using the concept of degree of membership. The characteristic function or membership function not only gives 0 or 1 but can also give values between 0 and 1.

A fuzzy set is a set with fuzzy boundaries. In fuzzy logic, the truth of any statement becomes a matter of degree. Any statement can be fuzzy. A value between 0 and 1 represents the degree of membership, also called membership value. The fuzzy boundaries can be considered according to the rules that are going to be used. As the number of rules increased, the degree of membership will become more accurate.

2.7 Membership Function

The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, it represents the degree of truth as an extension of valuation. Degrees of truth are often confused with probabilities, although they are conceptually distinct, because fuzzy truth represents membership in vaguely defined sets, not likelihood of some event or condition. [11]. There is a very wide selection to choose these membership function.

The simplest membership functions are formed using straight lines. Of these, the simplest is the triangular membership function, and it has the function name trimf. This function is nothing more than a collection of three points forming a triangle. The trapezoidal membership function, trapmf, has a flat top and really is just a truncated triangle curve. These straight line membership functions have the advantage of simplicity.

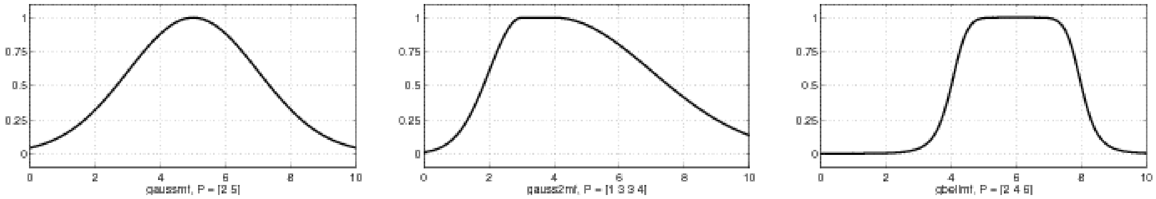


trimf trapmf

Figure 2.10 : Triangle and trapezoidal membership function

Two membership functions are built on the Gaussian distribution curve: a simple Gaussian curve and a two-sided composite of two different Gaussian curves. The two functions are gaussmf and gauss2mf.

The generalized bell membership function is specified by three parameters and has the function name gbellmf. The bell membership function has one more parameter than the Gaussian membership function, so it can approach a non-fuzzy set if the free parameter is tuned. Because of their smoothness and concise notation, Gaussian and bell membership functions are popular methods for specifying fuzzy sets. Both of these curves have the advantage of being smooth and nonzero at all points.



gaussmf gauss2mf gbellmf

Figure 2.11 : Gauss, gauss2 and generalized bell membership function

Although the Gaussian membership functions and bell membership functions achieve smoothness, they are unable to specify asymmetric membership functions, which are important in certain applications. The sigmoidal membership function, which is either open left or right. Asymmetric and closed membership functions can be synthesized using two sigmoidal functions, so in addition to the basic sigmf, it had the difference between two sigmoidal functions, dsigmf, and the product of two sigmoidal functions psigmf.

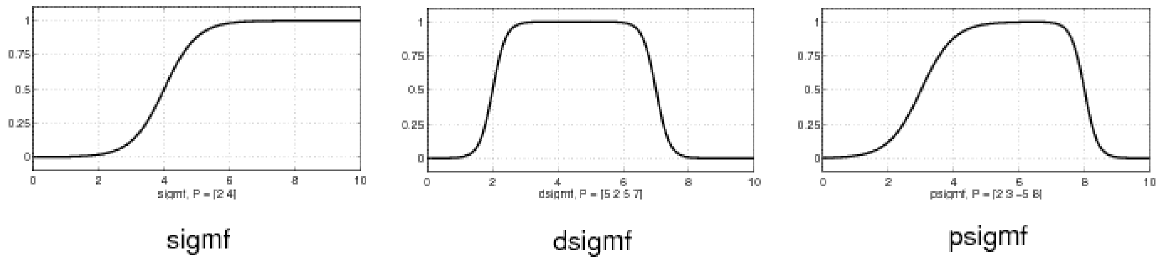


Figure 2.12 : Sigmoidal, dsi and psi membership function

Polynomial based curves account for several of the membership functions in the toolbox. Three related membership functions are the *Z*, *S*, and Pi curves, all named because of their shape. The function zmf is the asymmetrical polynomial curve open to the left, smf is the mirror-image function that opens to the right, and pimf is zero on both extremes with a rise in the middle.

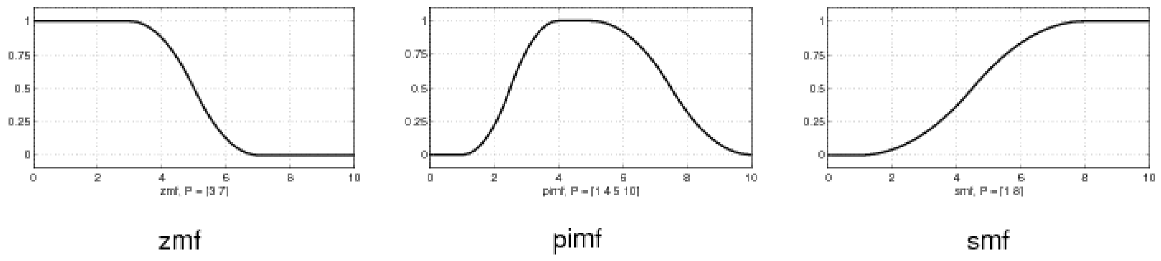
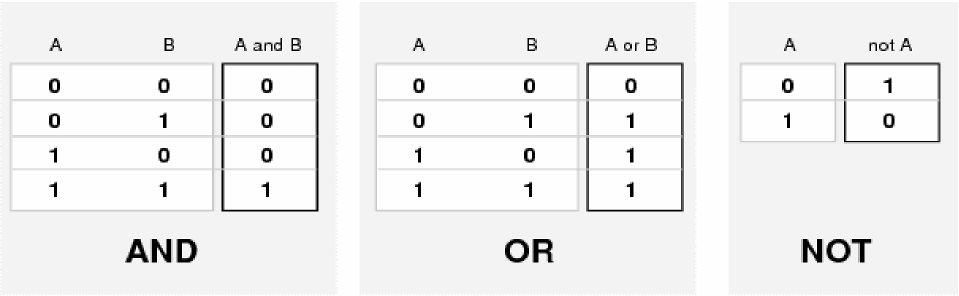


Figure 2.13 : Z shaped, Pi and S shaped membership function

2.8 Logical Operation

The most important thing to realize about fuzzy logical reasoning is the fact that it is a superset of standard Boolean logic. In other words, if the fuzzy values are kept at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold. As an example, consider the following standard truth tables.



A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	not A
0	1
1	0

NOT

Figure 2.14 : Truth tables for AND, OR and NOT

The next figure uses a graph to show the same information. In this figure, the truth table is converted to a plot of two fuzzy sets applied together to create one fuzzy set. The upper part of the figure displays plots corresponding to the preceding two-valued truth tables, while the lower part of the figure displays how the operations work over a continuously varying range of truth values *A* and *B* according to the fuzzy operations you have defined.

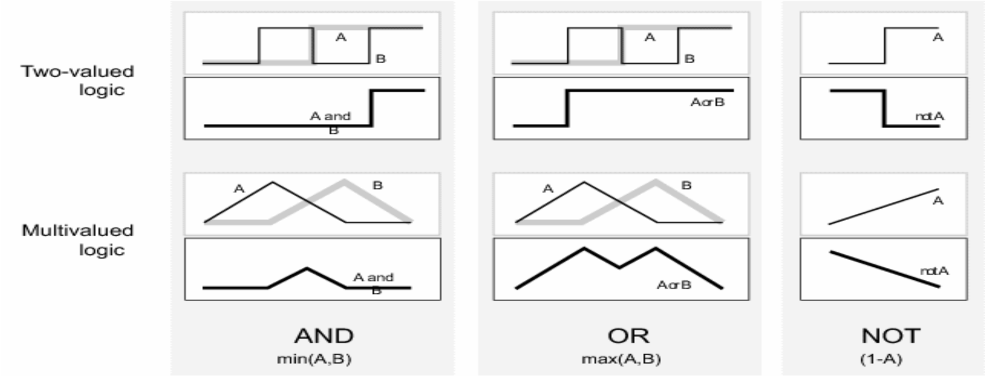


Figure 2.15 : The two-valued and multi valued logic for AND, OR and NOT

2.8.1 If- Then Rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. These if-then rule statements are used to formulate the conditional statements that comprise fuzzy logic.

IF-then rules is a three-part process.

i. **Fuzzify inputs**: Resolve all fuzzy statements in the antecedent to a degree of membership between 0 and 1. If there is only one part to the antecedent, then this is the degree of support for the rule.
ii. **Apply fuzzy operator to multiple part antecedents**: If there are multiple parts to the antecedent, apply fuzzy logic operators and resolve the antecedent to a single number between 0 and 1. This is the degree of support for the rule.
iii. **Apply implication method**: Use the degree of support for the entire rule to shape the output fuzzy set. The consequent of a fuzzy rule assigns an entire fuzzy set to the output.

iv. This fuzzy set is represented by a membership function that is chosen to indicate the qualities of the consequent. If the antecedent is only partially true, then the output fuzzy set is truncated according to the implication method.

Fuzzy logic also poses the ability to mimic the human mind to effectively employ modes of reasoning that are approximate rather than exact. C. Wen Hung et. al.[15] [14] used fuzzy approaches to determine optimal threshold values of fruit's images, and fuzzy degree of matching was applied to classify the color and size of fruit. This method has also been applied in [7][8].

2.9 Fuzzy Inference System

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be make. The process of fuzzy inference involves all of the pieces which are membership functions, fuzzy logic operators, and if-then rules. There are two types of fuzzy inferences systems that can be implemented in the Fuzzy Logic toolbox, Mamdani-type and Sugeno-type.

Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology. Mamdani – type inference, expects the output of membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification.

Mamdani-type inference

A type of fuzzy inference in which the fuzzy sets from the consequent of each rule are combined through the aggregation operator and the resulting fuzzy set is defuzzified to yield the output of the system.

Sugeno-Type Fuzzy Inference

Sugeno, or Takagi-Sugeno-Kang is a method of fuzzy inference. It is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

Advantages of the Sugeno Method :

- i. It is computationally efficient.
- ii. It works well with linear techniques (e.g., PID control).
- iii. It works well with optimization and adaptive techniques.
- iv. It has guaranteed continuity of the output surface.
- v. It is well suited to mathematical analysis.

Advantages of the Mamdani Method :

- i. It is intuitive.
- ii. It has widespread acceptance.
- iii. It is well suited to human input.

CHAPTER 3

METHODOLOGY

3.1 Operation Concept

This study is confined to the whole bunch of an oil palm fruit. The sample of this study are based on three categories of ripeness of fruits which are ripe, unripe and under ripe. The MATLAB software was used to perform analysis on the colour of the images.

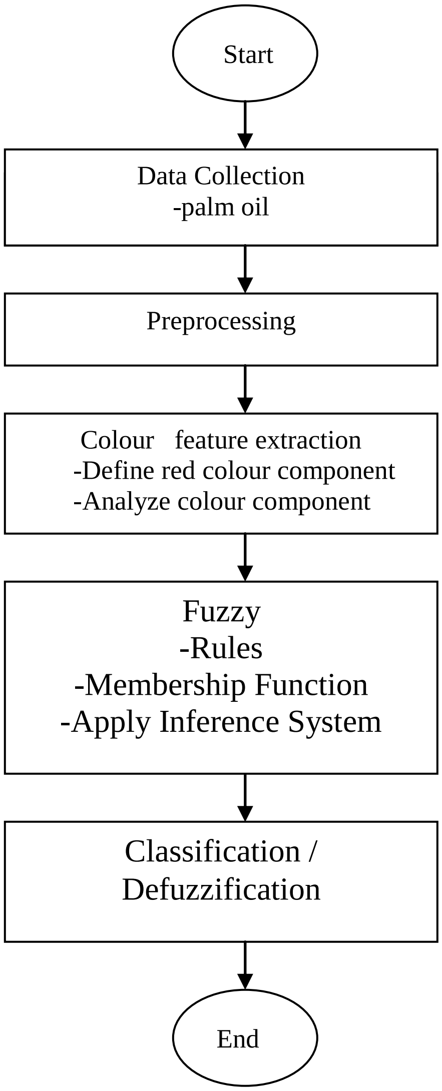


Figure 3.1: Palm Oil grading System Flow Chart

3.1.1 Data Collection

In this research, the sample image of palm oil are analyze and based on maturity of fruit which are ripe, under ripe, and unripe. All data collection are obtained from Dr Kamarul Hawari Ghazali from their previous research (RDU070357). The image are captured by using digital camera. The oil palm bunch colour is a main features to determine its category. The captured image is stored in RGB format with scale size 240x320 pixels. Analysis of pixel values is started by study the component of Red, Green and Blue in RGB image information. As fruit bunches is graded by looking at its colour red, analyze of RGB components will determine the intensity of red in each fruits category[2].

3.1.2 Pre-processing

The purpose of pre-processing is to improve the image data that suppressed undesired distortions or enhanced some important image features for further processing [14][16]. As manual grading by human are focusing on fruits color, the noise that disturb color features has to be removed. Color density is analyzed to determine the ripeness of each bunch. However the background color of the images may mislead the RGB color density of each bunch. For this reason, we need to work on the bunch without its background color. After remove the others color and change it to a black color as black return values 0 and don’t affect the next process of color filter.

Table 3.1, 3.2 and 3.3 shown that pixel value of RGB at sample image. This pixel value also referred to as the noise. From the pixel value maximum and minimum threshold value can be set for pass through the filtering process.

The figure 5.2, 5.3, 5.5, 5.6, 5.8 and 5.9 in chapter 5, show the original images and after filtering for each categories ripeness.

Table 3.1 : Pixel value of red colour

No	Ripe 1	Ripe 2	Ripe 3	Ripe 4	Ripe 5
1	252	243	147	138	193
2	213	234	248	171	233
3	134	102	246	207	233
4	113	216	116	255	255
5	147	241	193	246	221
6	169	160	162	243	212
7	229	209	151	202	199
8	250	210	123	233	209
9	176	197	201	213	68
10	206	238	227	209	203
11	214	210	82	179	97
12	254	191	189	220	214
13	173	254	135	234	255
14	91	195	87	185	215
15	255	247	176	212	168
16	144	203	80	190	185
17	197	255	142	229	158
18	183	155	212	198	176
19	245	246	228	207	211
20	233	242	166	249	113

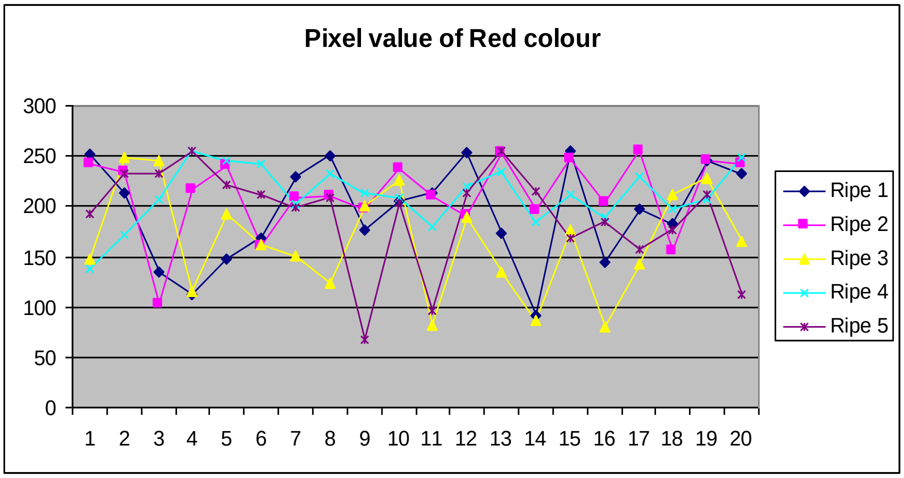


Figure 3.2 : Graph Plot of Red colour

Based on the above graph, as much as 20 times the pixel value are taken from the sample images. The maximum and minimum value is set right on the average frequency of each sample image. Thus the threshold value for the color red can be set right by its maximum value is 255 and the minimum value is 171.

Table 3.2 : Pixel value of green colour

No	Ripe 1	Ripe 2	Ripe 3	Ripe 4	Ripe 5
1	208	251	170	119	154
2	219	205	236	200	207
3	148	71	241	241	246
4	121	182	82	254	247
5	141	244	161	247	221
6	194	124	124	231	229
7	216	168	113	224	162
8	229	174	92	185	210
9	201	175	172	195	27
10	133	226	196	172	153
11	180	176	110	170	49
12	209	195	121	181	218
13	110	255	110	214	237
14	68	180	88	169	216
15	196	250	145	184	165
16	114	196	84	173	174
17	133	243	127	215	186
18	121	117	224	223	139
19	211	210	200	170	182
20	149	240	184	244	63

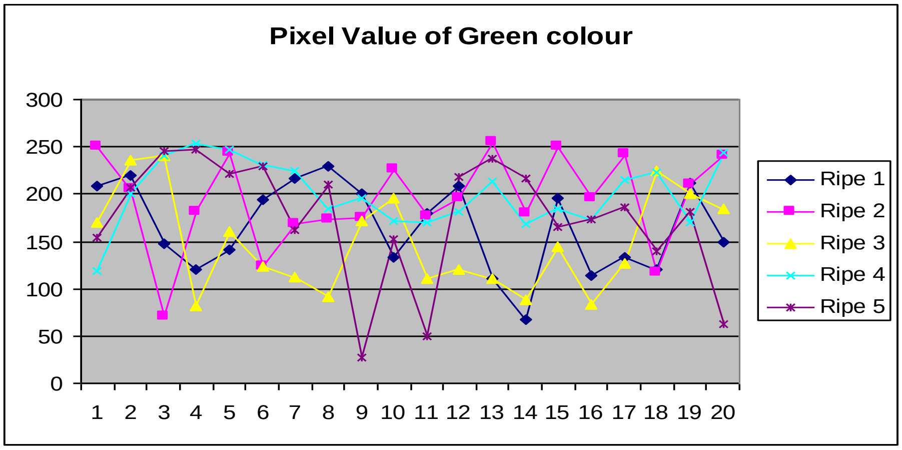


Figure 3.3 : Graph Plot of Green colour

Also for the green colour, based on the above graph, as much as 20 times the pixel value are taken from the sample images. Thus the threshold value for the color green colour are set right by its maximum value is 164 and the minimum value is 40. The parameter of grading oil palm are based on reddish. So, the main colour for every categories is red and green colour are not much effect for this grading.

Table 3.3 : Pixel value of blue colour

No	Ripe 1	Ripe 2	Ripe 3	Ripe 4	Ripe 5
1	208	251	92	121	113
2	219	205	214	92	180
3	148	71	209	146	156
4	121	182	57	235	251
5	141	244	150	233	147
6	194	124	103	173	159
7	216	168	94	115	136
8	229	174	74	173	150
9	201	175	112	180	0
10	133	226	168	127	126
11	180	176	69	105	13
12	209	195	48	122	159
13	110	255	44	155	219
14	68	180	48	144	114
15	196	250	78	160	88
16	114	196	61	127	129
17	133	243	42	176	86
18	121	117	100	121	94
19	211	210	116	126	148
20	149	240	124	188	40

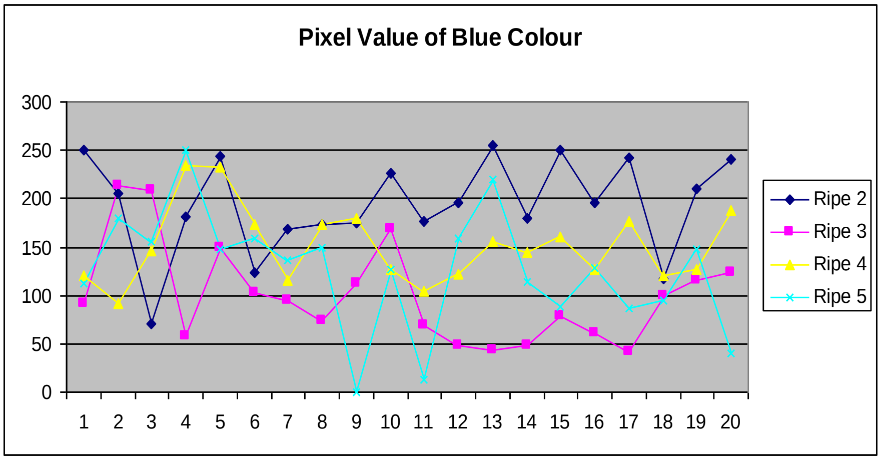
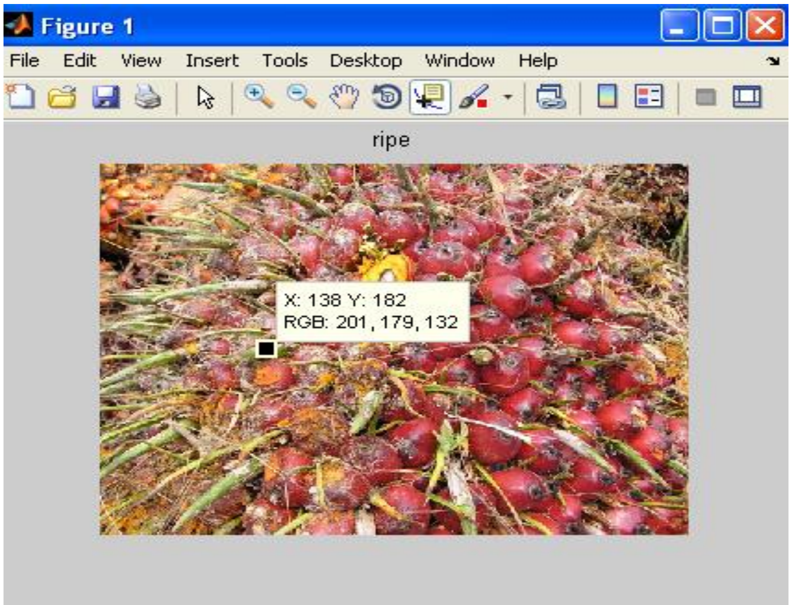


Figure 3.4 : Graph Plot of Blue colour

For blue colour as shown figure 3.4 above, this colour also not much useful in this grading. So the threshold value for the color blue colour are set right by its maximum value is 92 and the minimum value is 30. At the graph also can see the higher value are 255 and the lowers are 0, but the average frequency are taken. This mean not every sample image are had same value, so we need to choose average value.



X: 138 Y: 182
RGB: 201, 179, 132

Figure 3.5 : Read Pixel value of noise

3.1.3 Feature Extraction

The RGB color features are extracted after completion of the pre processing stage. The sum of each color component (Red, Green and Blue) was computed for every pixel area to get the mean value. The background of the image that provide the pixel value of 0 (black) was not included in the summation since the main focus is the RGB color of the oil palm rind.

In this feature extraction, masking technique are used after threshold technique. Masking is the technique of covering a stamped image so that other images may be placed partly over it without the overlapping area being visible. After that, the result will get reddish colour.

Equation (1), (2) and (3) are used to classify the oil palm bunch into their proper categories. A total of 15 pictures are used to calculate the mean or average, x and standard deviation,σ of the RGB bands.

Overipe interval = $X_{red/green/blue} \pm \sigma_{red/green/blue}$ (1)

Ripe interval = $X_{red/green/blue} \pm \sigma_{red/green/blue}$ (2)

Underipe interval = $X_{red/green/blue} \pm \sigma_{red/green/blue}$ (3)

Another calculation for the extraction of mean red, green and blue are, assume an RGB image sized m x n . The red, green and blue channels are R(x,y),G(x,y) and B(x,y), respectively, where $x \epsilon m$ and $y \epsilon n.x$.

The mean red component is given as

$$R_{mean} = \frac{\sum_{x-1}\sum_{y-1}F_{R1}(x,y)}{\sum_{x-1}\sum_{y-1}F_{R2}(x,y)} \qquad (4)$$

Where:

$$F_{R1}\,(x,y)= \begin{cases} R(x,y)\ , & R(x,y)>0 \\ 0 & , R(x,y)=0 \end{cases}$$

and:

$$F_{R2}\,(x,y)= \begin{cases} 1\ , & R(x,y)>0 \\ 0, & R(x,y\ )=0 \end{cases}$$

The mean green component is given as:

$$G_{mean} = \frac{\sum_{x-1}\sum_{y-1}F_{G1}(x,y)}{\sum_{x-1}\sum_{y-1}F_{G2}(x,y)} \qquad (5)$$

Where:

$$F_{G1}\,(x,y)= \begin{cases} G(x,y)\ , & G(x,y)>0 \\ 1 & , G(x,y\ )=0 \end{cases}$$

and:

$$F_{G2}\,(x,y)= \begin{cases} 1\ , & G(x,y)>0 \\ 0, & G(x,y\ )=0 \end{cases}$$

Finally, the mean blue component is given as:

$$B_{mean} = \frac{\sum_{x-1}\sum_{y-1}F_{B1}(x,y)}{\sum_{x-1}\sum_{y-1}F_{B2}(x,y)} \qquad (6)$$

Where:

$$F_{B1}(x,y) = \begin{cases} B(x,y), & B(x,y) > 0 \\ 0, & B(x,y) = 0 \end{cases}$$

and:

$$F_{B2}(x,y) = \begin{cases} 1, & B(x,y) > 0 \\ 0, & B(x,y) = 0 \end{cases}$$

3.1.4 Recognition

As mentioned, fuzzy logic system was developed to simulate human decision making in determining ripeness of palm oil bunch based on the color . The palm oil is classified into three levels of ripeness which is unripe, ripe and under ripe. Three of the mean color component (red, green and blue) are identified as the input parameters to the fuzzy system .

It’s about 21 rules that had been made as shown in figure 3.4 below, but only 7 are use. This is because the pattern for the classify are red component must be higher than green and blue, and green component must be high than blue. This pattern are obtain from the pixel value at the image also threshold pattern at program for filtering image samples. So that only 7 rules are accepted as shown in table 3.5. The output parameter for this system is the ripeness level of the palm oil.

Red	**Green**	**Blue**	**Category**
Large	Small	Small	Ripe
Large	Small	Medium	Ripe
Large	Small	Large	Ripe
Large	Medium	Small	Ripe
Large	Medium	Medium	Ripe
Large	Medium	Large	Ripe
Large	Large	Small	Ripe
Large	Large	Medium	Ripe
Large	Large	Large	Ripe
Medium	Small	Small	Under ripe
Medium	Small	Medium	Under ripe
Medium	Small	Large	Under ripe
Medium	Medium	Small	Under ripe
Medium	Medium	Medium	Under ripe
Medium	Medium	Large	Under ripe
Medium	Large	Small	Under ripe
Medium	Large	Medium	Under ripe
Medium	Large	Large	Under ripe
Small	Small	Small	Unripe
Small	Small	Medium	Unripe
Small	Small	Large	Unripe
Small	Medium	Small	Unripe
Small	Medium	Medium	Unripe
Small	Medium	Large	Unripe
Small	Large	Small	Unripe
Small	Large	Medium	Unripe
Small	Large	Large	Unripe

Table 3.4: Fuzzy Logic Rules

Below is 7 rules Fuzzy Logic that are used in this research :

Red	Green	Blue	Category
Large	Small	Small	Ripe
Large	Medium	Small	Ripe
Large	Medium	Medium	Ripe
Medium	Small	Small	Under ripe
Medium	Medium	Small	Under ripe
Small	Small	Small	Unripe
Small	Medium	Medium	Unripe

Table 3.5: 7 Fuzzy Logic Rules Selected

A lot of testing are done before chosen this 7 rules. This is to ensure that rules are made are correct and appropriate to the chosen membership function. Error in make rules give a big effect to system which can not be classified by category of oil palm fruit. Indirectly, this makes the system did not work. Thus accuracy is essential, and run tests to get good rules.

If-Then Rules

1. If Red is Large And Green is Small And Blue is Small Then Category is Ripe.
2. If Red is Large And Green is Medium And Blue is Small Then Category is Ripe.
3. If Red is Large And Green is Medium And Blue is Medium Then Category is Ripe.
4. If Red is Medium And Green is small And Blue is Small Then Category is Under Ripe.
5. If Red is Medium And Green is Medium And Blue is Small Then Category is Under Ripe.
6. If Red is Small And Green is Small And Blue is Small Then Category is Unripe.
7. If Red is Small And Green is Medium And Blue is Medium Then Category is Unripe

3.1.5 Membership Function

The system will be tested by using seven rules with three inputs and one output. This seven rules system also will be tested by using different eleven types of membership function. But only several that suitable for accepted to do a testing and give a result as shown in table 5.4 in chapter 5 result and discussion. For verification purpose, more rules system will be tested in order to find the more accurate system that can be obtain by adding the numbers of rules instead considering another membership function.

After done a testing triangle and trapezoidal membership function are chosen for this system fuzzy logic because it more accuracy than others. This happen because, the shape trapezoidal and triangle are more sharp and give reading value at range more accurately.

3.1.6 Fuzzy Logic

Introduction To Fuzzy Logic Toolbox

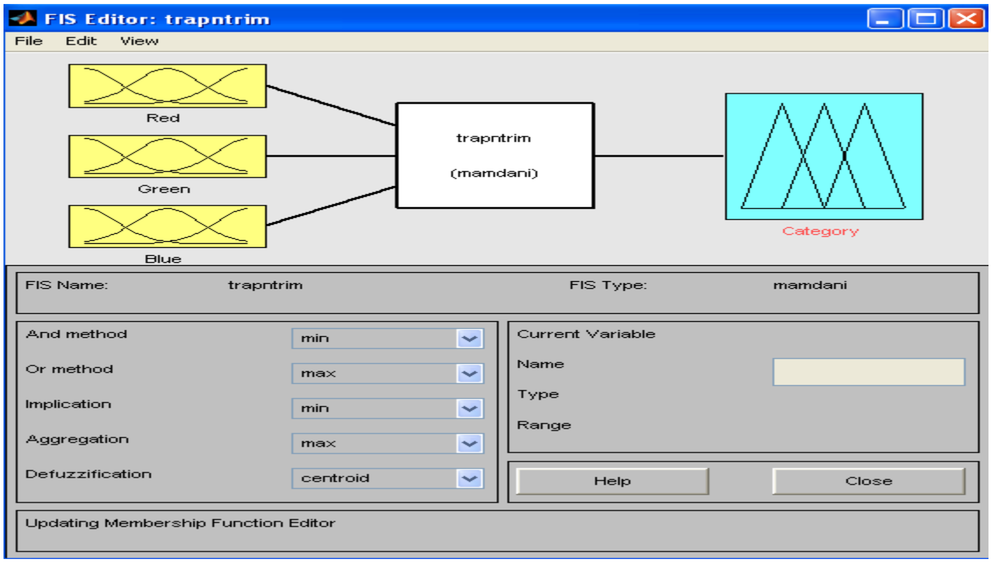
Fuzzy Logic Toolbox allow several things to be done, but the most important things are to be a place where a fuzzy inference system can be created or edited. These systems can be created by using graphical tool.

Fuzzy Inferences System (FIS)

The most important thing for build Fuzzy Inferences System (FIS) is the rules and membership must be right. In this fuzzy logic system, 3 input and 1 output has been used.Mean value of RGB colour are the input for fuzzy logic and the output are ripeness of oil palm bunch ripe, under ripe and unripe.

i. Define the structure of Inferences System

This part are for set the input, output and rules for fuzzy logic system as shown in figure 3.6. In this system the input are red, green and blue which are RGB mean value.



FIS Name: trapntrim

FIS Type: mamdani

And method: min

Or method: max

Implication: min

Aggregation: max

Defuzzification: centroid

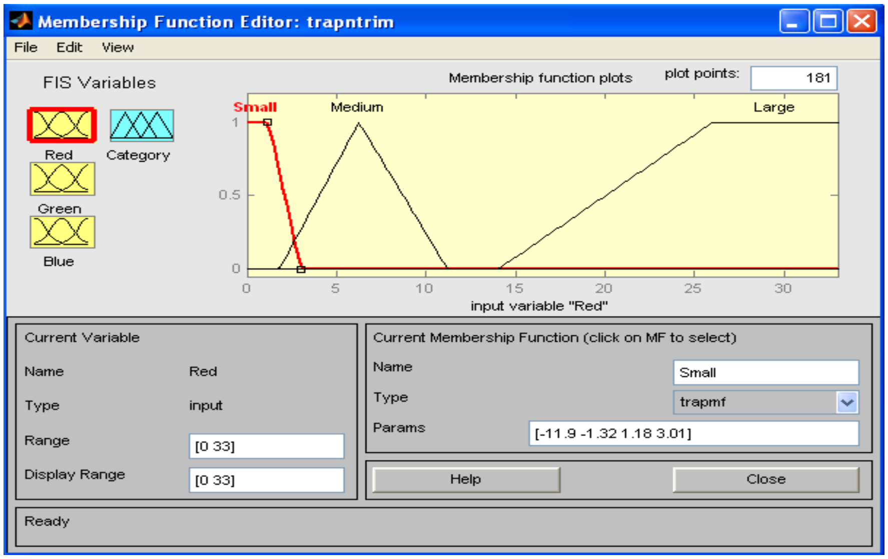
Updating Membership Function Editor

Figure 3.6 : FIS Editor

ii. Define the Membership Functions

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. In this project the input are red, green and blue with range 0 to 33 and the output is category with range also from 0 to 33. This range are refer to the mean value of image after the feature extraction.

After done some testing for choosing membership function, at the last trapezoidal and triangle membership function are chosen. Figure 3.7, 3.8, 3.9 shown the membership function that has been choose.



Current Variable

Name Red

Type input

Range [0 33]

Display Range [0 33]

Current Membership Function (click on MF to select)

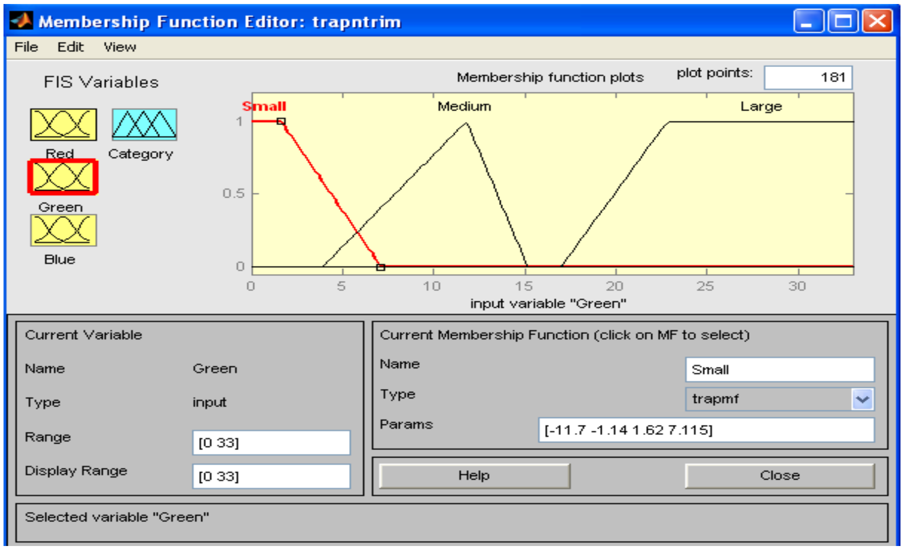
Name Small

Type trapmf

Params [-11.9 -1.32 1.18 3.01]

Ready

Figure 3.7 : Membership Function for input Red



Current Variable

Name Green

Type input

Range [0 33]

Display Range [0 33]

Current Membership Function (click on MF to select)

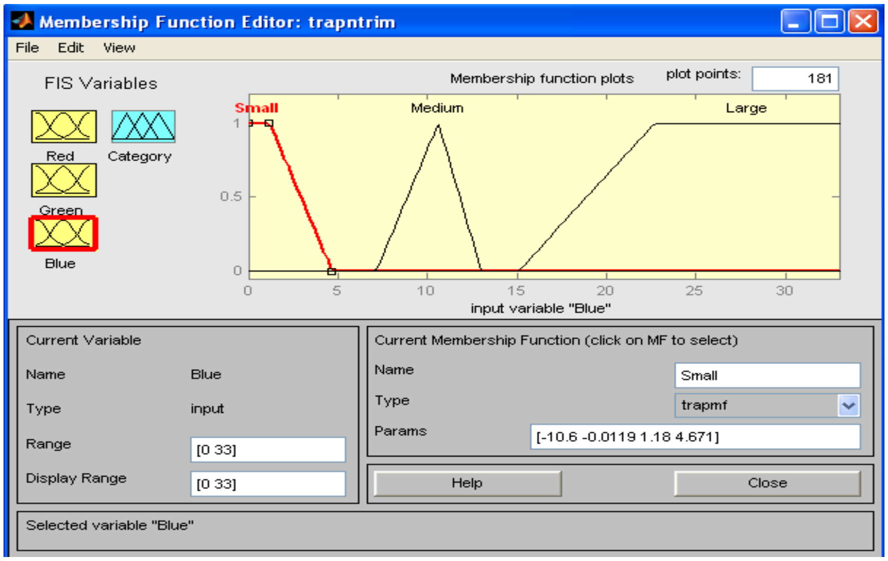
Name Small

Type trapmf

Params [-11.7 -1.14 1.62 7.115]

Selected variable "Green"

Figure 3.8 : Membership Function for input Green



plot points: 181

Current Variable

Name: Blue

Type: input

Range: [0 33]

Display Range: [0 33]

Current Membership Function (click on MF to select)

Name: Small

Type: trapmf

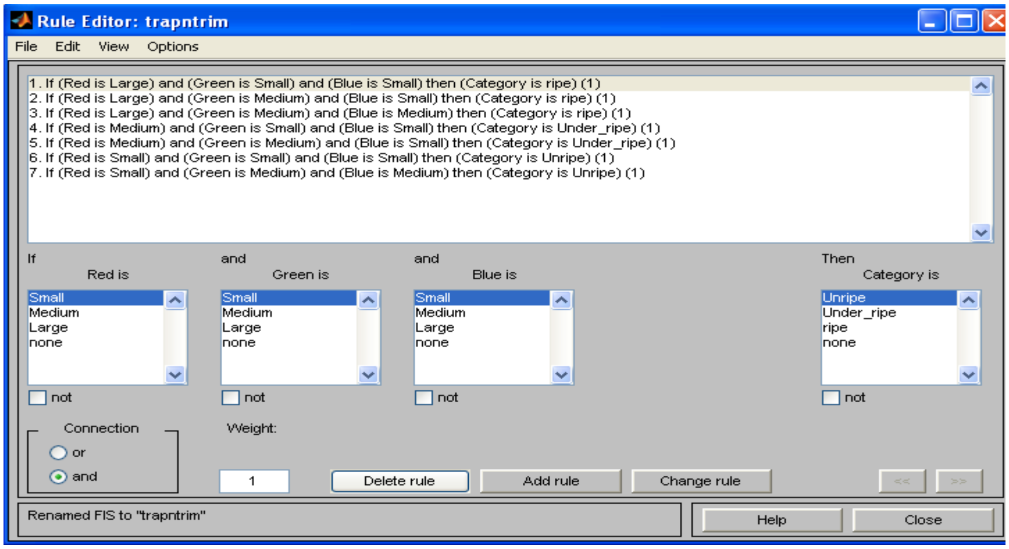
Params: [-10.6 -0.0119 1.18 4.671]

Selected variable "Blue"

Figure 3.9 : Membership Function for input Blue

iii. Construct IF-THEN Rules

The rules must be construct properly, so that rules and membership function are suitable and give the correct output. Figure 3.10 shown the rules that has made for this fuzzy logic system.



1. If (Red is Large) and (Green is Small) and (Blue is Small) then (Category is ripe) (1)
2. If (Red is Large) and (Green is Medium) and (Blue is Small) then (Category is ripe) (1)
3. If (Red is Large) and (Green is Medium) and (Blue is Medium) then (Category is ripe) (1)
4. If (Red is Medium) and (Green is Small) and (Blue is Small) then (Category is Under_ripe) (1)
5. If (Red is Medium) and (Green is Medium) and (Blue is Small) then (Category is Under_ripe) (1)
6. If (Red is Small) and (Green is Small) and (Blue is Small) then (Category is Unripe) (1)
7. If (Red is Small) and (Green is Medium) and (Blue is Medium) then (Category is Unripe) (1)

Red is: Small; Green is: Small; Blue is: Small; Category is: Unripe

Connection: ○ or ● and

Weight: 1

Renamed FIS to "trapntrim"

Figure 3.10 : IF-THEN Rule

iv. Visualize and Deploy the Fuzzy Inference Diagram

Since decisions are based on the testing of all of the rules in a Fuzzy Inference System, the rule must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the output of each rule combined into a single fuzzy set.

3.1.7 Defuzzification

The input for the defuzzification process is a fuzzy set and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. There are five built-in methods supported: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum.

Fuzzy logic will make a classification from the mean value of palm oil, and maturity will be determined.

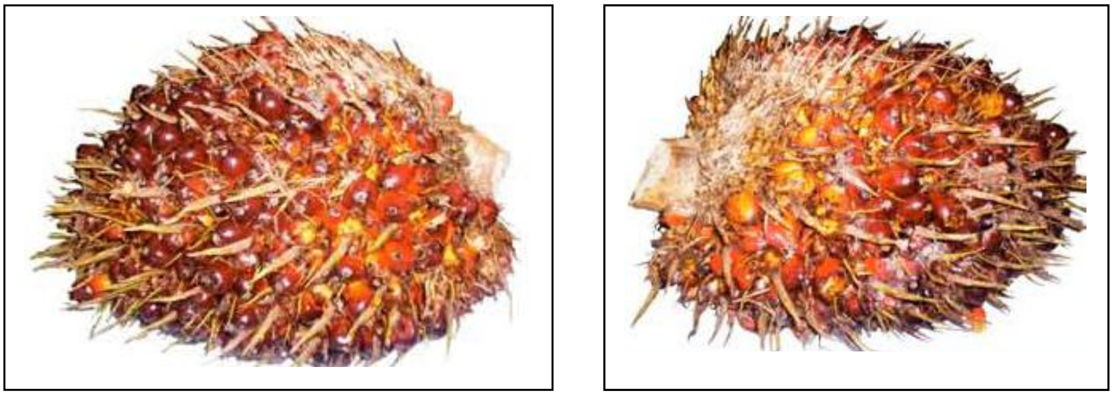


Figure 3.11 :The two images for ripe bunch

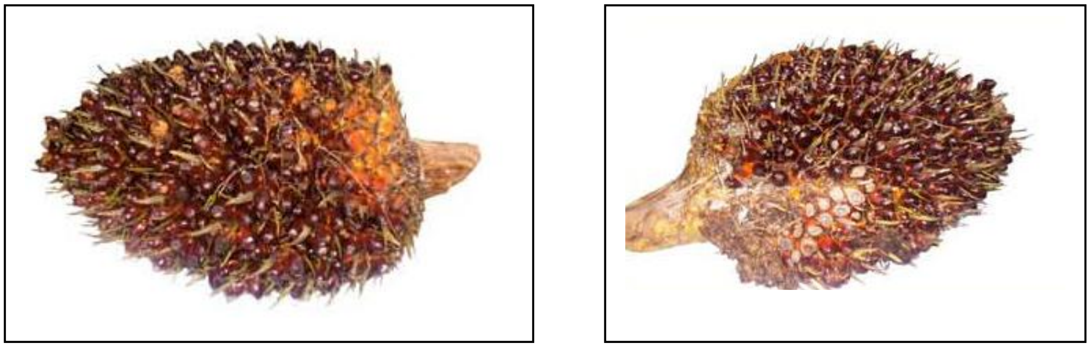


Figure 3.12 : The two images for under ripe bunch

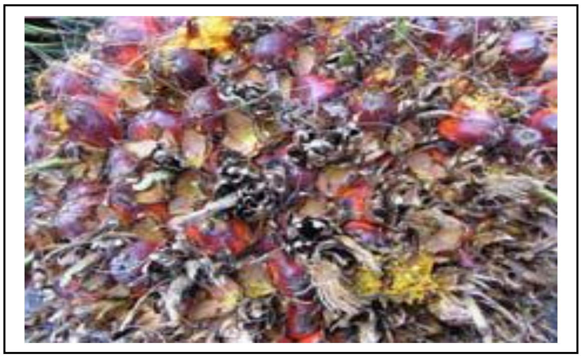
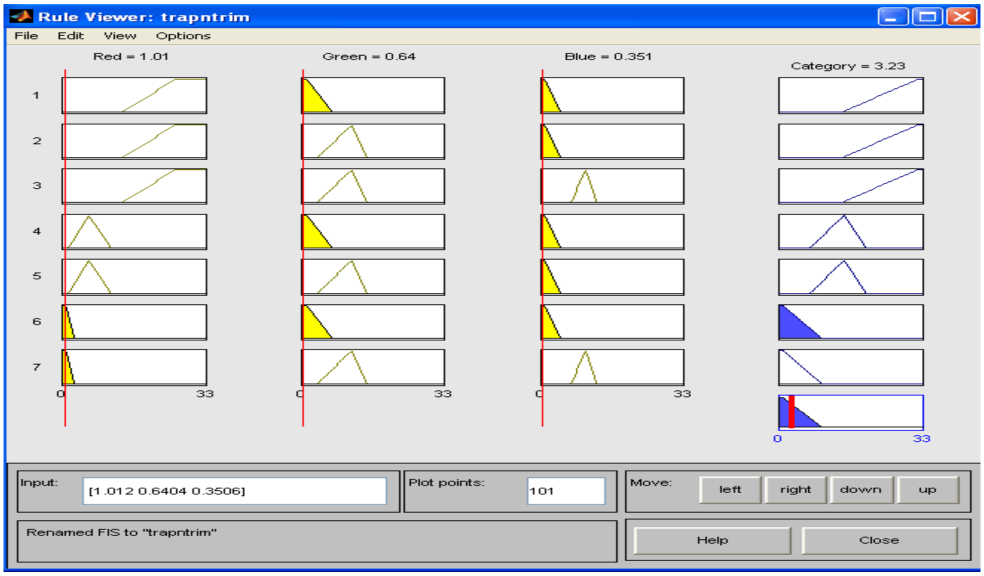


Figure 3.13 : The image for unripe bunch

Figure 3.14 below shown the defuzzification of the fuzzy logic system in unripe category. Detail about this fuzzification refer to result and discussion in chapter 5.



Input: [1.012 0.6404 0.3506]

Plot points: 101

Renamed FIS to "trapntrim"

Figure 3.14: Defuzzification result that obtained from Rule Viewer

CHAPTER 4

DEVELOPMENT SOFTWARE PROGRAM

4.1 Overall Flow Of Software

This chapter are consist several step for build this software program. Start from develop programming for image processing and using fuzzy logic to make a decision and lastly by Graphical User Interface GUI to show the result.

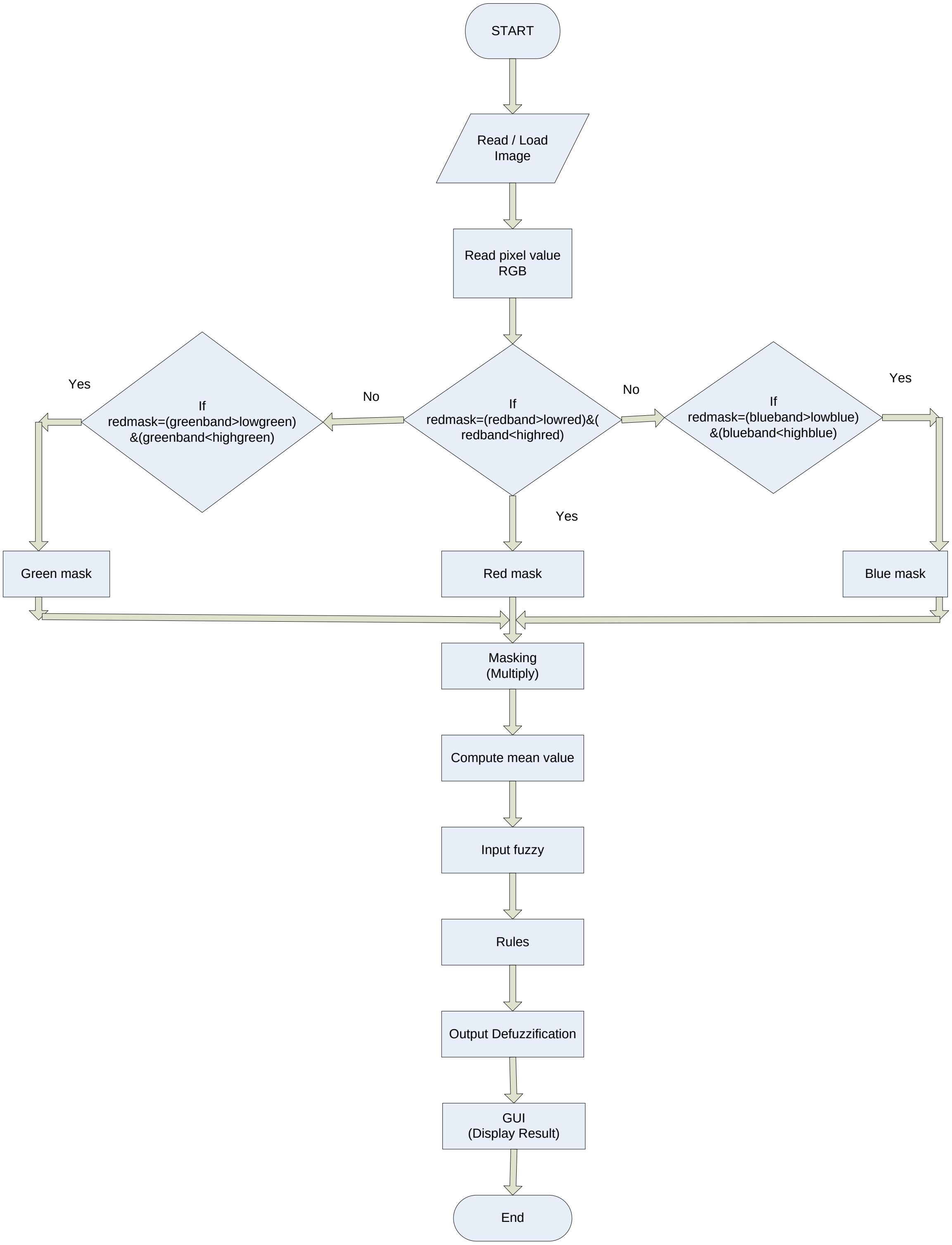


Figure 4.1 :Overall flow of software program

4.2 M-File

The software part of the project was produced by using Image processing technique and fuzzy logic in MATLAB software. In this MATLAB software there was the essential tool used throughout the whole software program development progress. The software program is written in the M-file in MATLAB software as shown in Figure 4.2. There are used some tools in MATLAB desktop such as Command Window, Command History Window and Workspace Browser.

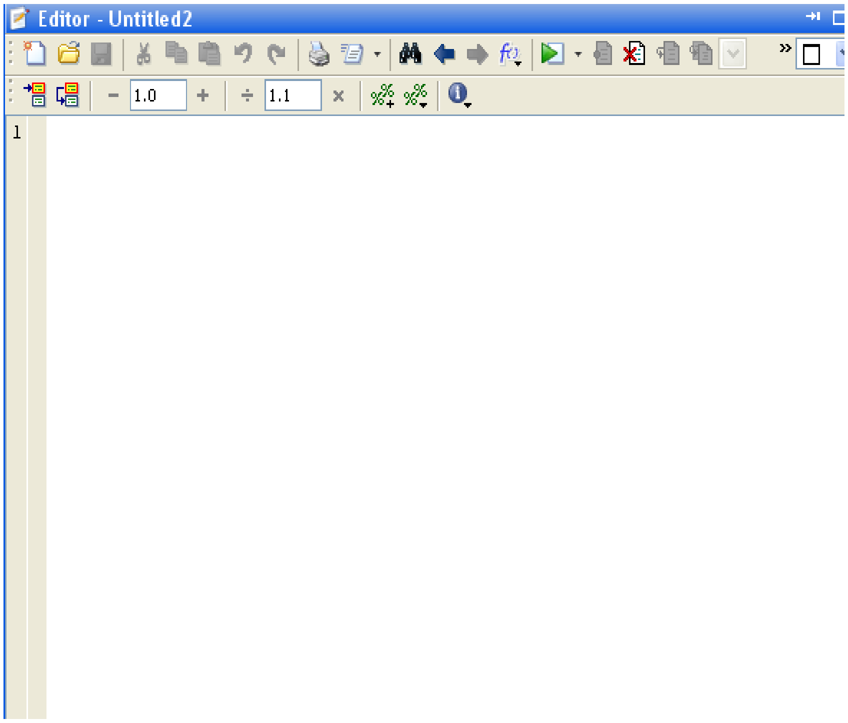
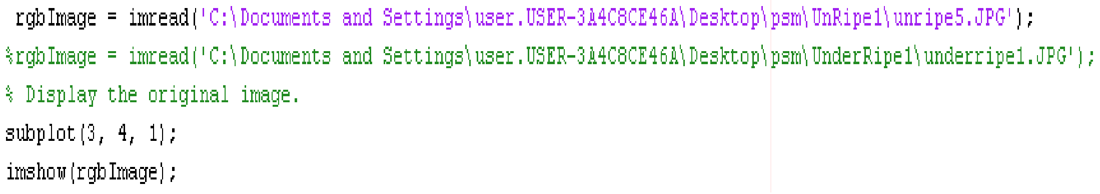


Figure 4.2 : New blank M-File

4.2.1 Read Image / Load Image

As shown in figure 4.3 below, first step in this software program are to import or read an image. After inserting the image the next step can be proceed. Most of the images are saving in following format which is supported by MATLAB:

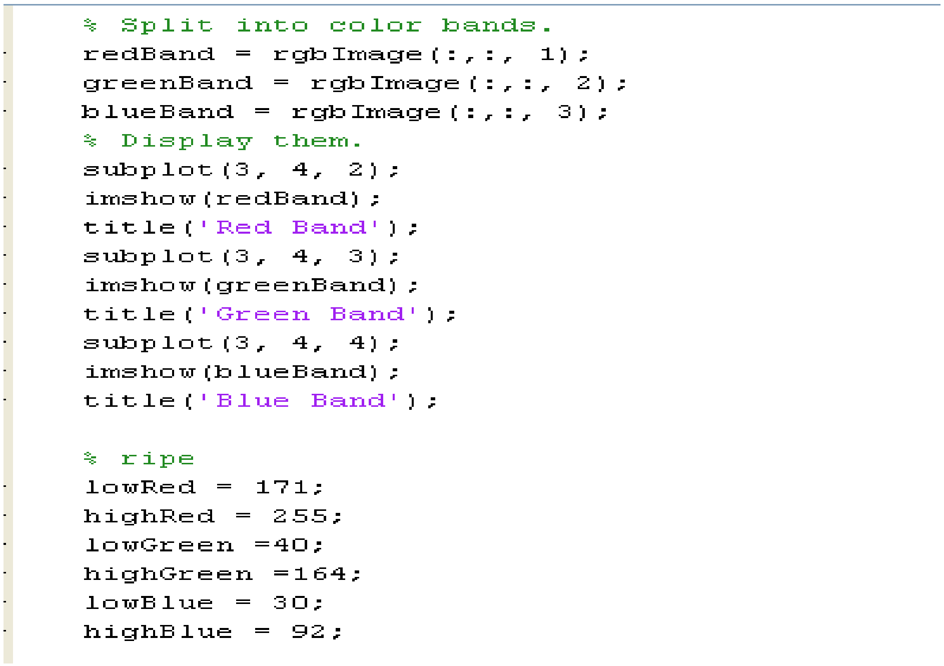
1. BMP
2. HDF
3. JPEG
4. PCX
5. TIFF
6. XWB



```
 rgbImage = imread('C:\Documents and Settings\user.USER-3A4C8CE46A\Desktop\psm\UnRipe1\unripe5.JPG');
%rgbImage = imread('C:\Documents and Settings\user.USER-3A4C8CE46A\Desktop\psm\UnderRipe1\underripe1.JPG');
% Display the original image.
subplot(3, 4, 1);
imshow(rgbImage);
```

Figure 4.3 : Programming for read the image

4.2.2 Read Pixel Value



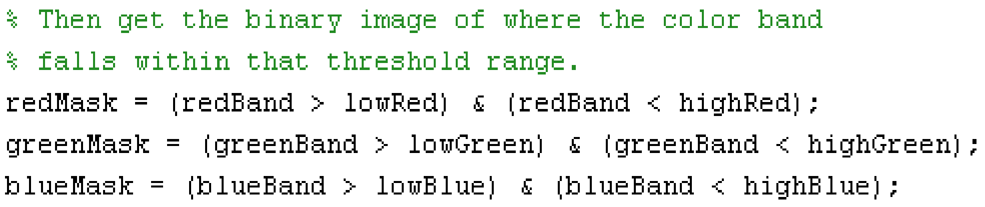
```
% Split into color bands.
redBand = rgbImage(:,:, 1);
greenBand = rgbImage(:,:, 2);
blueBand = rgbImage(:,:, 3);
% Display them.
subplot(3, 4, 2);
imshow(redBand);
title('Red Band');
subplot(3, 4, 3);
imshow(greenBand);
title('Green Band');
subplot(3, 4, 4);
imshow(blueBand);
title('Blue Band');

% ripe
lowRed = 171;
highRed = 255;
lowGreen =40;
highGreen =164;
lowBlue = 30;
highBlue = 92;
```

Figure 4.4 : Set the pixel value for threshold

The value of low and high red, green and blue are set based on pixel value. All category are used same value. This is because for ripe category, the result are more to reddish and unripe are opposite of it. So by using the same value the result for unripe category are more to black because unripe category less of reddish colour. For under ripe, the colour of oil palm bunch are not much different from unripe, so its also use the same range value.

4.2.3 Threshold



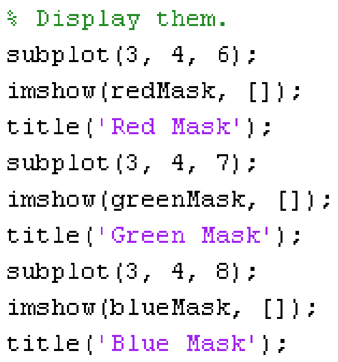
```
% Then get the binary image of where the color band
% falls within that threshold range.
redMask = (redBand > lowRed) & (redBand < highRed);
greenMask = (greenBand > lowGreen) & (greenBand < highGreen);
blueMask = (blueBand > lowBlue) & (blueBand < highBlue);
```

Figure 4.5 : Range of Threshold

Accuracy or efficiency of the software program will be tested from the range maximum and minimum that has formulate. Randomly change the pixel value of noise at programming to get the more accurate value of mean of RGB colour.

The threshold range are set up based on pattern at features of oil palm bunch. Parameter for grading oil palm also based on reddish. So the red colour must be greater than green blue.

4.2.4 Red, Green and Blue Mask

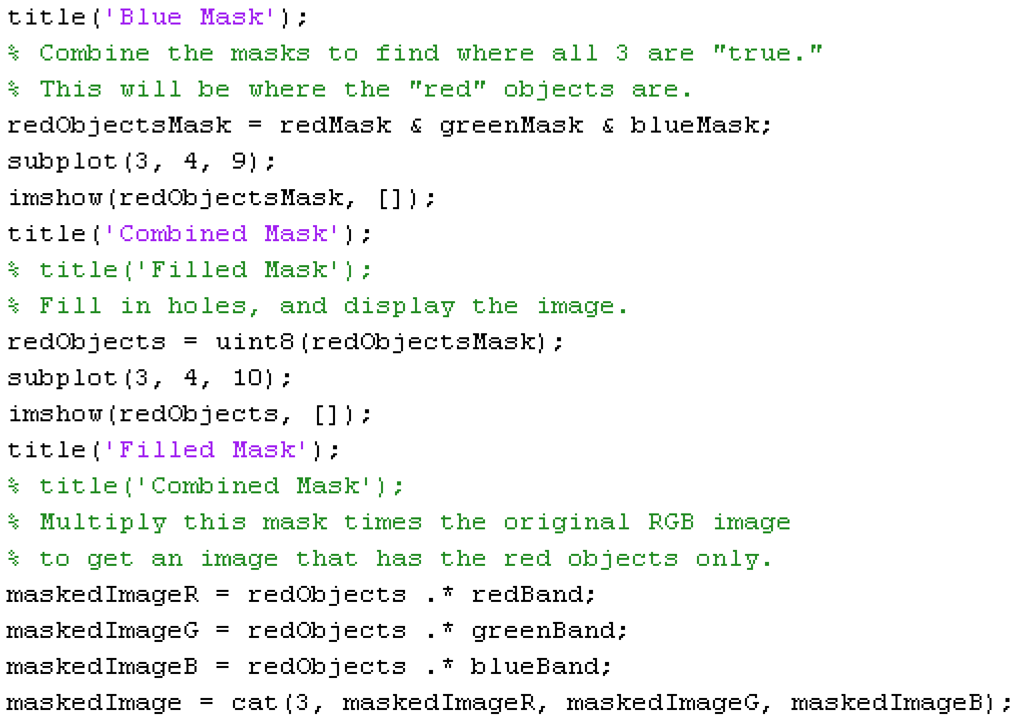


```
% Display them.
subplot(3, 4, 6);
imshow(redMask, []);
title('Red Mask');
subplot(3, 4, 7);
imshow(greenMask, []);
title('Green Mask');
subplot(3, 4, 8);
imshow(blueMask, []);
title('Blue Mask');
```

Figure 4.6 : Red, Green and Blue Mask

After threshold process, every component are display based on their colour. This process separate the colour red, green and blue from the colour band and the format image are in grayscale. It look like make a hole, because the red, and blue colour are in 1,1,1 pixel value background in black 0,0,0.

4.2.5 Masking (Multiply)

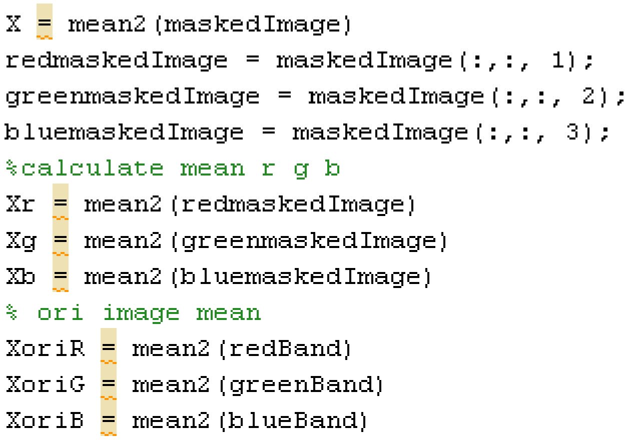


```
title('Blue Mask');
% Combine the masks to find where all 3 are "true."
% This will be where the "red" objects are.
redObjectsMask = redMask & greenMask & blueMask;
subplot(3, 4, 9);
imshow(redObjectsMask, []);
title('Combined Mask');
% title('Filled Mask');
% Fill in holes, and display the image.
redObjects = uint8(redObjectsMask);
subplot(3, 4, 10);
imshow(redObjects, []);
title('Filled Mask');
% title('Combined Mask');
% Multiply this mask times the original RGB image
% to get an image that has the red objects only.
maskedImageR = redObjects .* redBand;
maskedImageG = redObjects .* greenBand;
maskedImageB = redObjects .* blueBand;
maskedImage = cat(3, maskedImageR, maskedImageG, maskedImageB);
```

Figure 4.7 : Masking

Combination of red, green and blue mask for get the red objects mask. To get an image that had the red objects only, multiply the red, green and blue mask with original image. After that by using cat technique, concatenate with maskedImage R, G and B will get masked image.

4.2.6 Compute mean value of RGB



```
X = mean2(maskedImage)
redmaskedImage = maskedImage(:,:, 1);
greenmaskedImage = maskedImage(:,:, 2);
bluemaskedImage = maskedImage(:,:, 3);
%calculate mean r g b
Xr = mean2(redmaskedImage)
Xg = mean2(greenmaskedImage)
Xb = mean2(bluemaskedImage)
% ori image mean
XoriR = mean2(redBand)
XoriG = mean2(greenBand)
XoriB = mean2(blueBand)
```

Figure 4.8 : Compute Mean Value

The end of this program is to compute the mean value for every masked image red, green and blue and also mean value of original image. The mean value for this program as show in table 5.1 , 5.2 ,5.3 at chapter 5.

4.3 Graphical User Interface (GUI)

A graphical user interface is a pictorial interface to a program. A good GUI can make programs easier to use providing the with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders and menus. The GUI should behave in an understandable and predictable manner, so that user knows what to expect when GUI perform an action.

A tool guide are need to allow a user to layout of GUI, selecting and aligning the components to be placed it. Figure 40 below shown the basic components of GUI. The large gray area with grid lines is the layout area. The layout editor window is a palette of GUI components along the left-hand side of the layout area.

In order to perform the running process of the software program in a more presentable and user friendly way, GUI is created. This GUI provides a step which are complete to run the whole program flow to configure a fuzzy logic and get a result for every category.

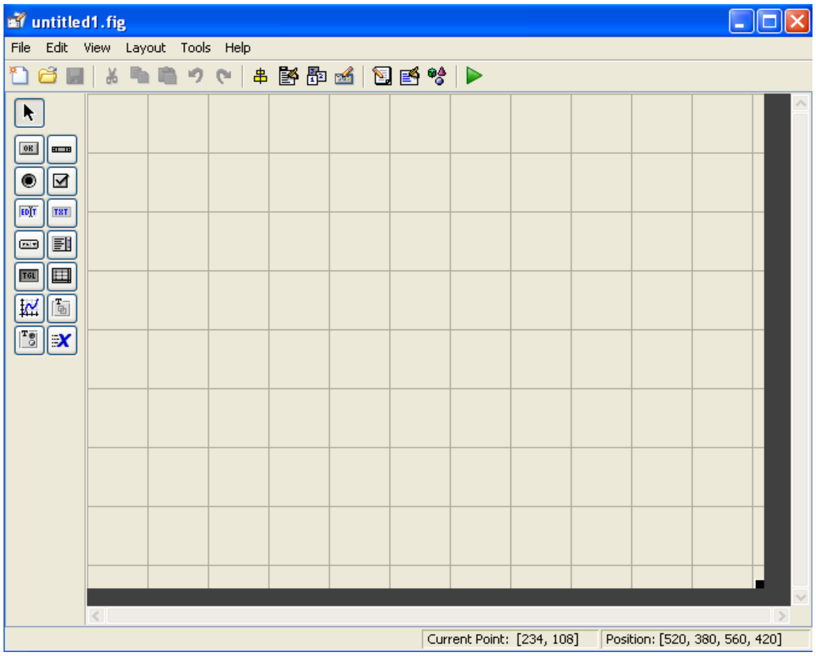

untitled1.fig
File Edit View Layout Tools Help
Current Point: [234, 108]
Position: [520, 380, 560, 420]

Figure 4.9 : The guide tool window

The GUI M-file that GUIDE generates is a function file. The name of the main function is the same as the name of the M-file. Each callback in the file is sub function of the main function.

When GUIDE generates an M-file, it automatically includes templates for the most commonly used callbacks for each component. GUIDE automatically includes two callbacks, the opening function and the output function, in every GUI M-file it creates.

This GUI developed for make this program system more systematic. GUI will display the result of ripeness after defuzzification. Using MATLAB tool called guide to lay out the components on a figure.

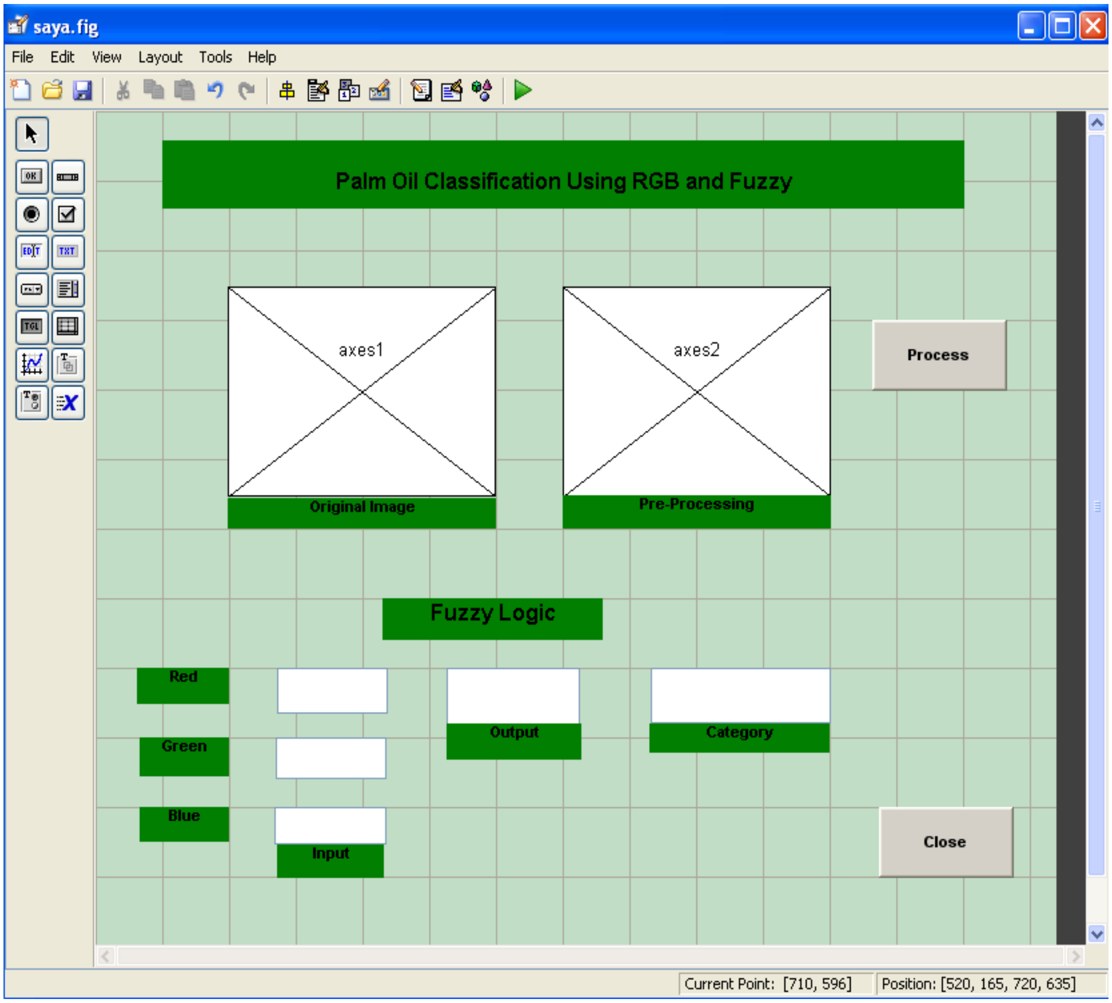

saya.fig
File Edit View Layout Tools Help
Palm Oil Classification Using RGB and Fuzzy
axes1
axes2
Process
Original Image
Pre-Processing
Fuzzy Logic
Red
Green
Blue
Input
Output
Category
Close
Current Point: [710, 596]
Position: [520, 165, 720, 635]

Figure 4.10 : GUI layout of Palm Oil Fruit Grading System using Fuzzy

CHAPTER 5

RESULT AND DISCUSSION

5.1 Introduction

This chapter consists of the discussions on the results from MATLAB programming**.** Some experiments had been conducted for the project. The experiment is conducted to find out mean value of RGB colour of maturity oil palm fruit bunches. Then, mean value of data collection is done by developed the software. The mean value of data collection use as input for fuzzy logic. An analysis on the ripeness of oil palm bunches of the system are made.

5.2 The Analysis Of Results

5.2.1 RGB Colour Extraction

The programming from MATLAB has calculated the mean colour intensity to find ripeness of oil palm bunches. This method is done by 15 pictures for samples, where there represents 5 samples for each category ripe, under ripe and ripe. The mean RGB for each category of oil palm fruit bunch sample is shown in table 7,8,9.

Table 5.1 : Mean RGB Intensity of the sample for ripe category

Mean of RGB			
Samples	Red	Green	Blue
1	20.3093	9.7414	7.0951
2	32.3004	13.4108	10.0139
3	27.0085	10.9087	8.2309
4	16.4028	7.4880	5.4829
5	14.0150	6.8584	4.8996

Table 5.2 : Mean RGB Intensity of the sample for under ripe category

Mean of RGB			
Samples	Red	Green	Blue
1	4.4255	2.4834	1.6167
2	7.7221	4.0798	2.6179
3	5.8406	2.9817	2.0627
4	6.0209	2.9249	2.1232
5	3.5088	1.9045	1.1818

Table 5.3 : Mean RGB Intensity of the sample for unripe category

Mean of RGB			
Samples	Red	Green	Blue
1	1.9587	1.3211	0.7086
2	0.7113	0.4715	0.2802
3	1.3811	0.6760	0.4869
4	0.7113	0.4715	0.2802
5	1.0123	0.6404	0.3506

The table clearly show that red is dominant colour for every ripeness categories. Meanwhile the others colour are noise and they are not required. As the content of red is a main features to distinguish between categories. All data in table 5.1, 5.2 and 5.3 shows the mean of red can be used to classify ripe, under ripe and unripe. Mean of red for ripe bunches are between 14.0150 and 32.3004, for under ripe and unripe categories mean value of red are between 3.5088, 7.7221 and 0.7113,1.9587. From that analysis, the feature of mean examined by using fuzzy logic system developed for automatic classification.

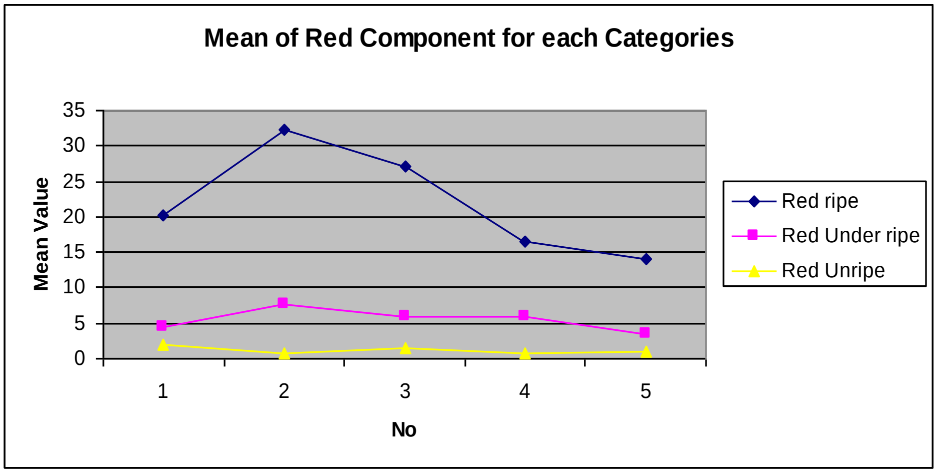


Figure 5.1: Mean of Red Component for each Categories

As shown in figure 5.1, the mean value of ripe category is discriminate from under ripe and unripe category. It show that in ripe category, red component had large value in sample image compared to unripe and under ripe. Although the mean of under ripe and unripe are had no much different of their red component but it still shown that red for under ripe are greater than red unripe. So, the distinguish of these categories, it easy to classify the oil palm categories. In the real application, it is enough for human grader to distinguish between these categories which are ripe, under ripe or unripe. The result that obtained is similar to the human grader where ripe and unripe category are two different group[2].

Ripe Category

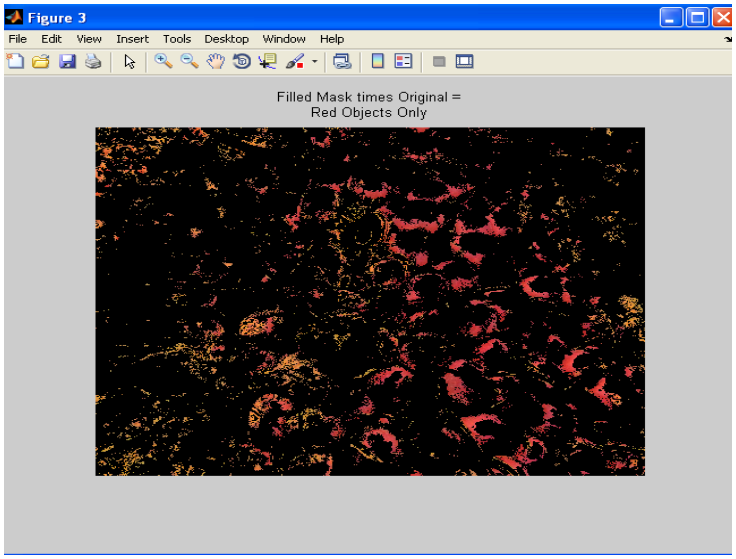


Figure 3
File Edit View Insert Tools Desktop Window Help
Filled Mask times Original =
Red Objects Only

Figure 5.2 : Result after feature extraction for ripe category

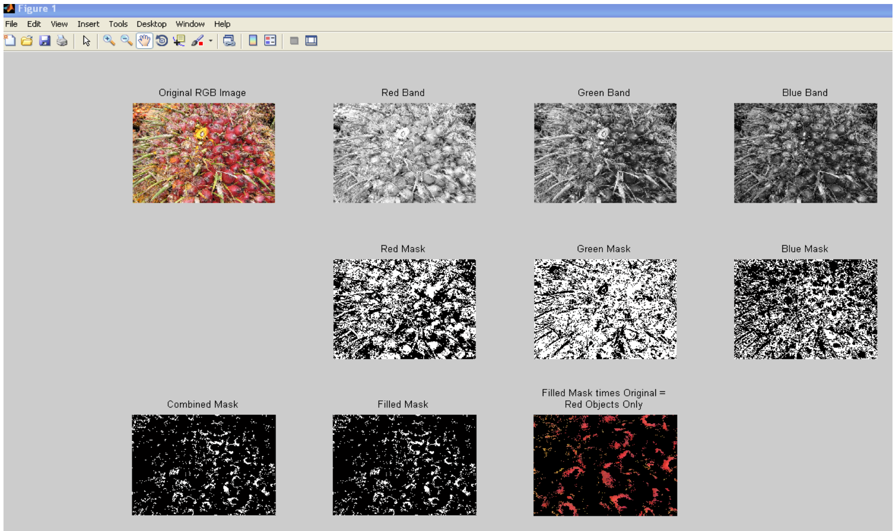


Figure 5.3 : Ripe category

The figure 5.2 and 5.3 show the output image from ripe categories. The image is in RGB image and go trough with the masking process to obtain the red colour. From the manual grading by “Malaysia Palm Oil Board (MPOB)” ripe oil palm bunches has red and orange fruit mesocarp orange outer layer.

This bunch has at least ten (10) socket lost fresh fruit and more than fifty percent (50%) were still stuck in the bunch[18]

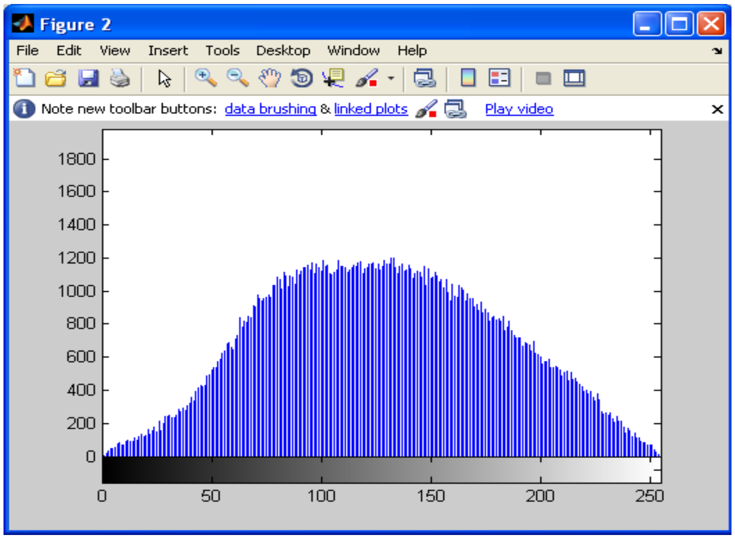


Figure 5.4 : Histogram original sample for ripe category

Figure 5.4 above shown the histogram for ripe category. This category had a lot of reddish colour. So the graph plot at the middle of 90 to 180 are high. This category are less of black colour.

Under ripe Category

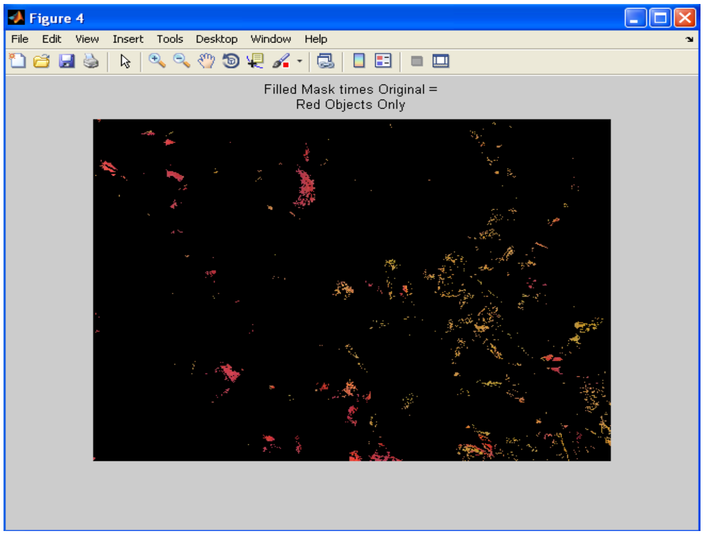


Figure 4
File Edit View Insert Tools Desktop Window Help
Filled Mask times Original =
Red Objects Only

Figure 5.5 : Result after feature extraction for under ripe category

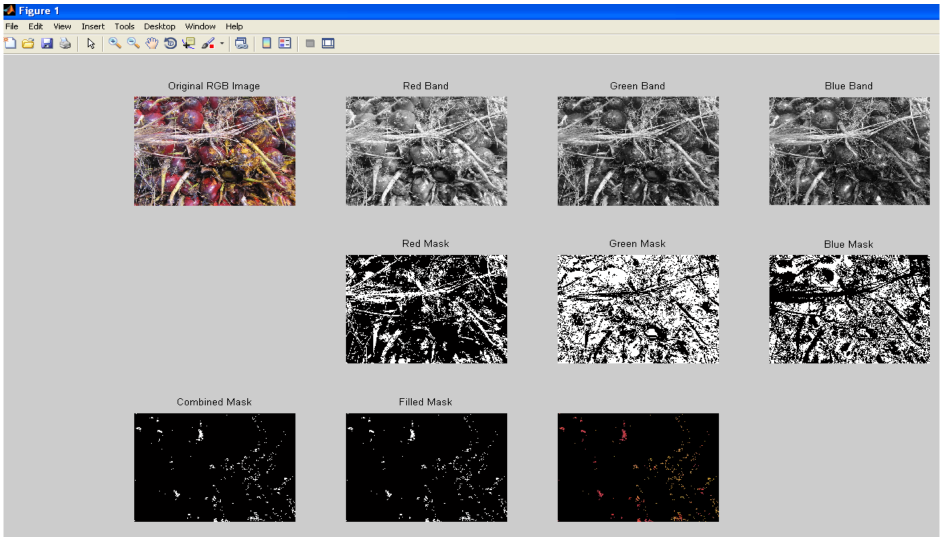


Figure 5.6 : Under ripe category

The reddish mesocarp of under ripe and unripe is almost similar to each other. The figure 5.5, 5.6, 5.8 and 5.9 clearly show that is difficult to differentiate between these two group categories. Under ripe oil palm bunches are reddish orange or reddish purple and the outer layer of the fruit mesocarp slightly orange in color. This bunch have less than ten (10) relay sockets fresh fruit during inspection at the factory[18].

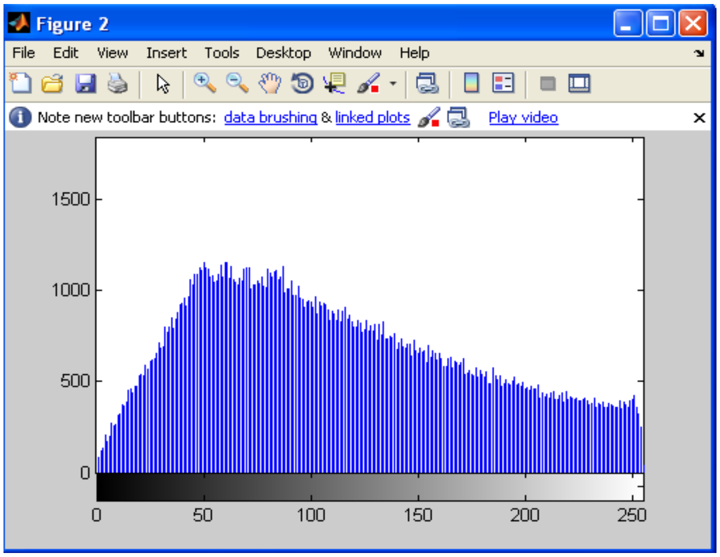


Figure 2
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
1500
1000
500
0
0
50
100
150
200
250

Figure 5.7 : Histogram original sample for Under ripe category

The figure 5.7 shown the histogram of under ripe category. It shows the distribution of grey levels in the image. The plot are more to 100 and below. This is because features of oil palm bunch under ripe category are less of red colour. So it more to 0.

Unripe Category

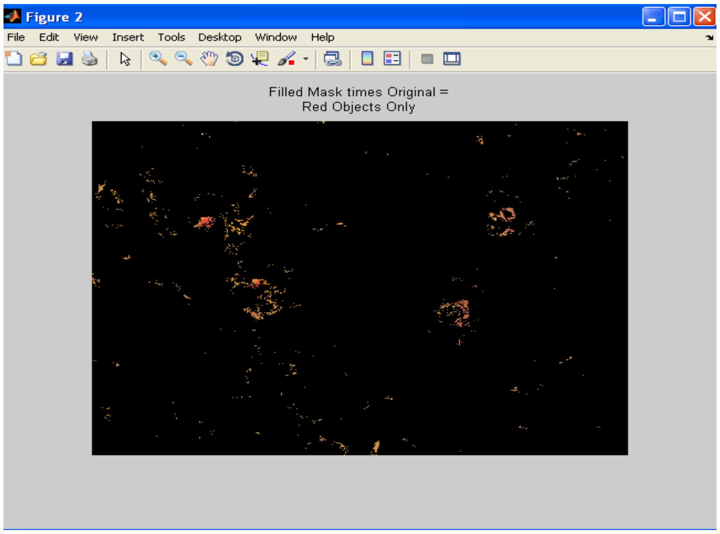


Figure 2
File Edit View Insert Tools Desktop Window Help
Filled Mask times Original =
Red Objects Only

Figure 5.8 : Result after feature extraction for unripe category

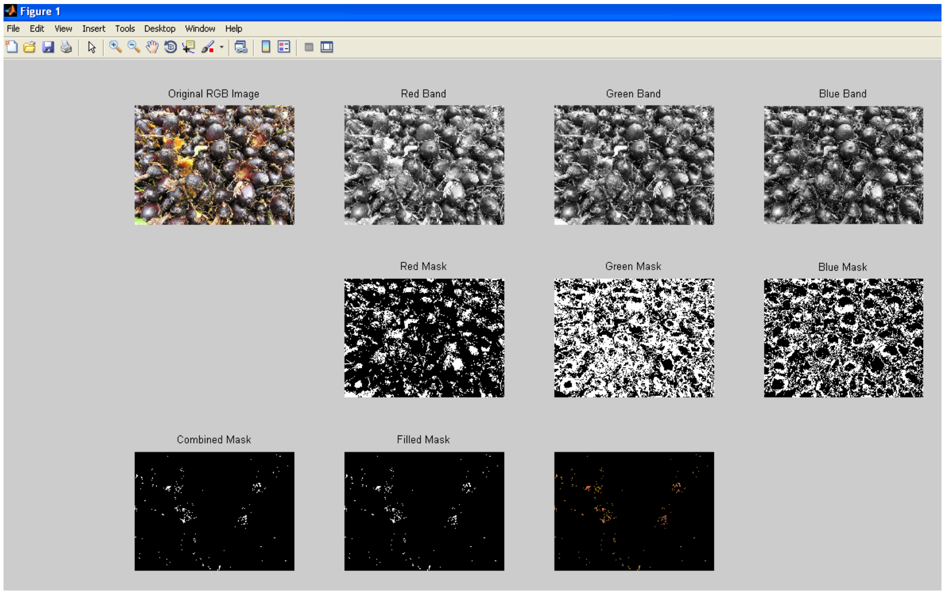


Figure 5.9 : Unripe category

Oil palm bunch under unripe category are look like more black and less reddish. Unripe bunches had black or dark purple and their mesocarp are yellowish at outer layer. This bunch does not had fresh fruit relay socket

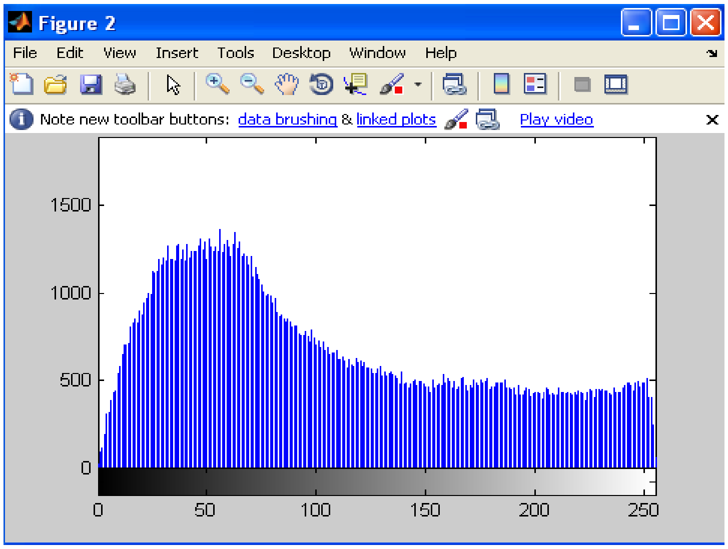


Figure 5.10 : Histogram original sample of Unripe category

The histogram in figure 5.10 above shown the histogram of unripe category where the plot had more to 0. As mention before unripe oil palm bunch are more to black, so the plot at 0 to 80 are high.

From the analysis that had make from the mean value and histogram, it show that every category had their own features to distinguish them. Its easy to grading of oil palm bunch category by using this method.

5.2.2 Membership Function Testing Result

Category	Membership Function	Membership Function	Accepted	Accuracy
Ripe	Triangle	Triangle	Yes	90%
	Trapezoidal	Trapezoidal	Yes	90%
	Triangle	Trapezoidal	Yes	90%
	Gbell	Gbell	No	40%
	Gbell	Gauss2	No	35%
	Gauss2	Gauss2	No	25%
Under ripe	Triangle	Triangle	Yes	60%
	Trapezoidal	Trapezoidal	Yes	65%
	Triangle	Trapezoidal	Yes	80%
	Gbell	Gbell	No	30%
	Gbell	Gauss2	No	35%
	Gauss2	Gauss2	No	30%
Unripe	Triangle	Triangle	Yes	90%
	Trapezoidal	Trapezoidal	Yes	90%
	Triangle	Trapepoidal	Yes	100%
	Gbell	Gbell	No	45%
	Gbell	Gauss2	No	30%
	Gauss2	Gauss2	No	20%

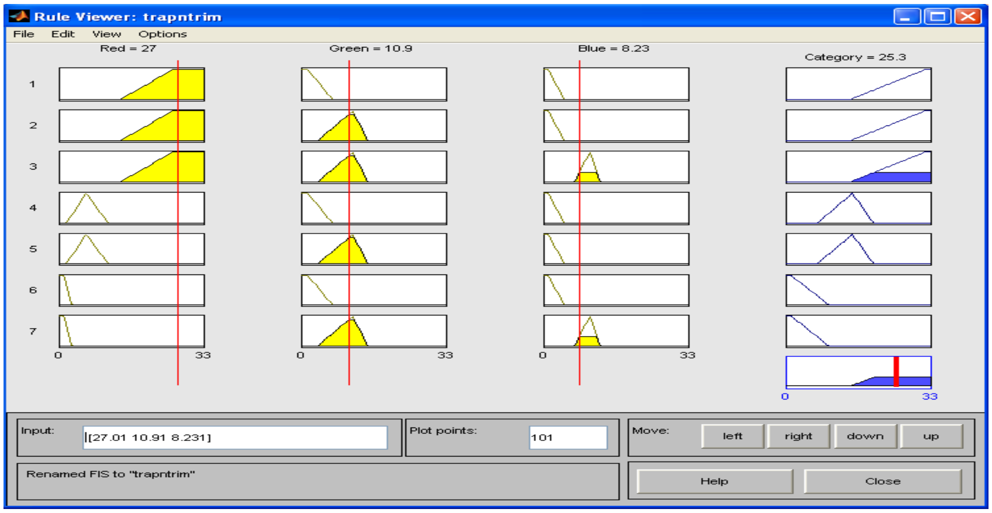
Table 5.4: Accuracy that membership function that been tested

As shown in table 5.4 above, the most accuracy membership function are triangle and trapezoidal which in 90% in ripe category, 80% in under ripe and 100% in unripe category. It had a shape which give result more accurate and easy to adjustable. After done this testing, the membership function that had been choose are trapezoidal and triangle. This two combination give a 0% in False Accepted Rate (FAR) and 10% in False Rejection Rate (FRR) in ripe category, for under ripe and unripe category the result are 0% and 0% for False Accepted Rate (FAR) for False Rejection Rate (FRR) 20% and 0%. Refer in figure 24,25,26 at chapter methodology software development where the part define membership function.

The crispy set that used are 0 to 255 refer to RGB range colour, but this is not suitable, because a lot of misclassification especially in under ripe and ripe category. So, the range of mean value RGB after feature extraction are used from 0 to 33.

5.2.3 Fuzzy Logic Decision

Results of fuzzy logic system are evaluated against the human graders to measure accuracy. It shows that the automated fuzzy logic grading system achieved 90% accuracy in grading ripe categories, 80% in grading under ripe categories and 100% in grading unripe categories. Figure 5.11, 5.12, 5.13 below shown the result from rule viewer.

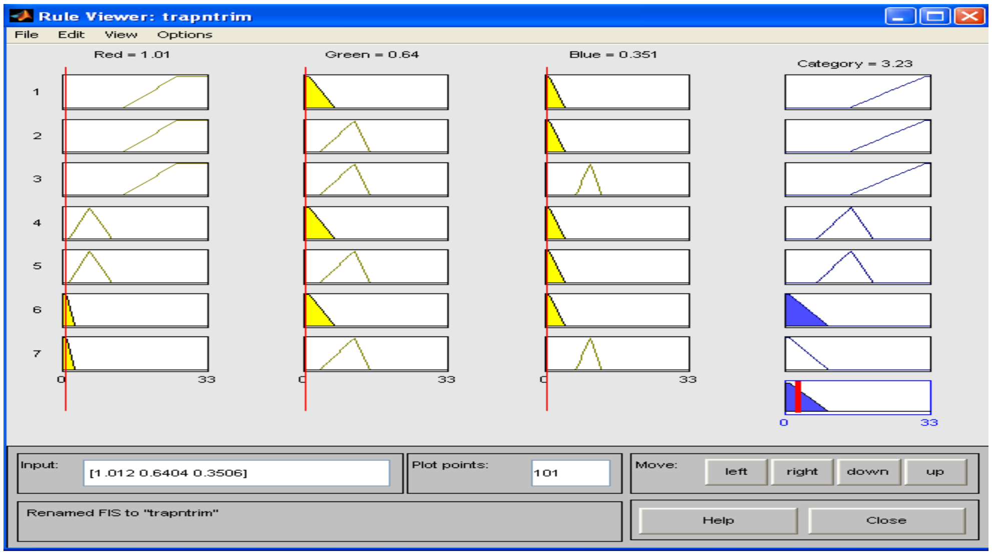


Input: [27.01 10.91 8.231]

Plot points: 101

Renamed FIS to "trapntrim"

Figure 5.11 : Ripe Defuzzification

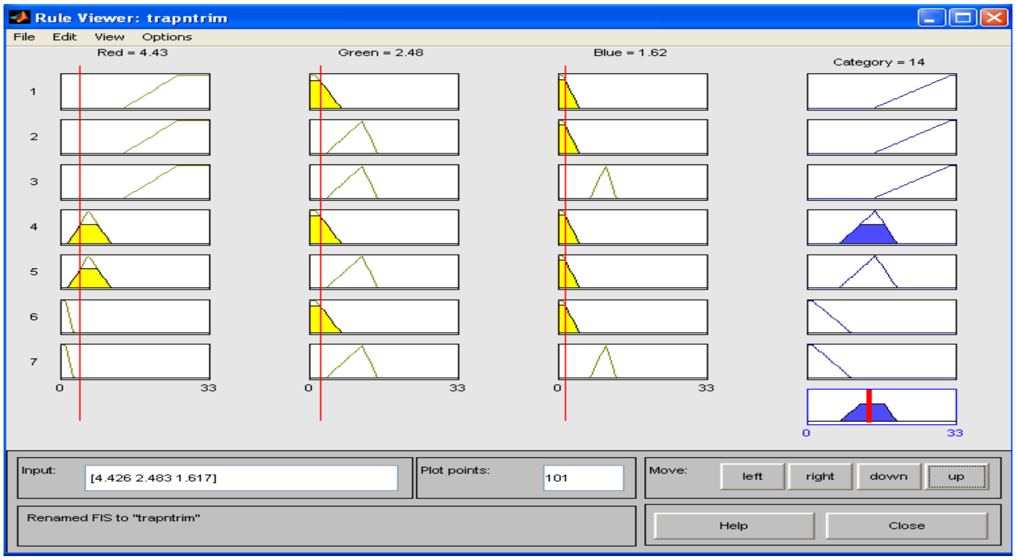


Input: [1.012 0.6404 0.3506]

Plot points: 101

Renamed FIS to "trapntrim"

Figure 5.12 : Unripe Defuzzification



Input: [4.426 2.483 1.617]

Plot points: 101

Renamed FIS to "trapntrim"

Figure 5.13 : Under ripe Defuzzification

Some misclassification is due to the design of the membership function and fuzzy rules that are based on experience and knowledge. Determination of membership functions in term of shape and boundary has a clear effect on the result of classification performed by fuzzy logic[19].

The parameters of fuzzy logic such threshold, which is to determine the overlapping amount and condition among the membership functions, input and output levels, and function rules must be tested for optimum classification.

5.3 Result GUI

By press the result button process, GUI will display the image original and after pe-processing, input and output fuzzy and category of oil palm bunch.

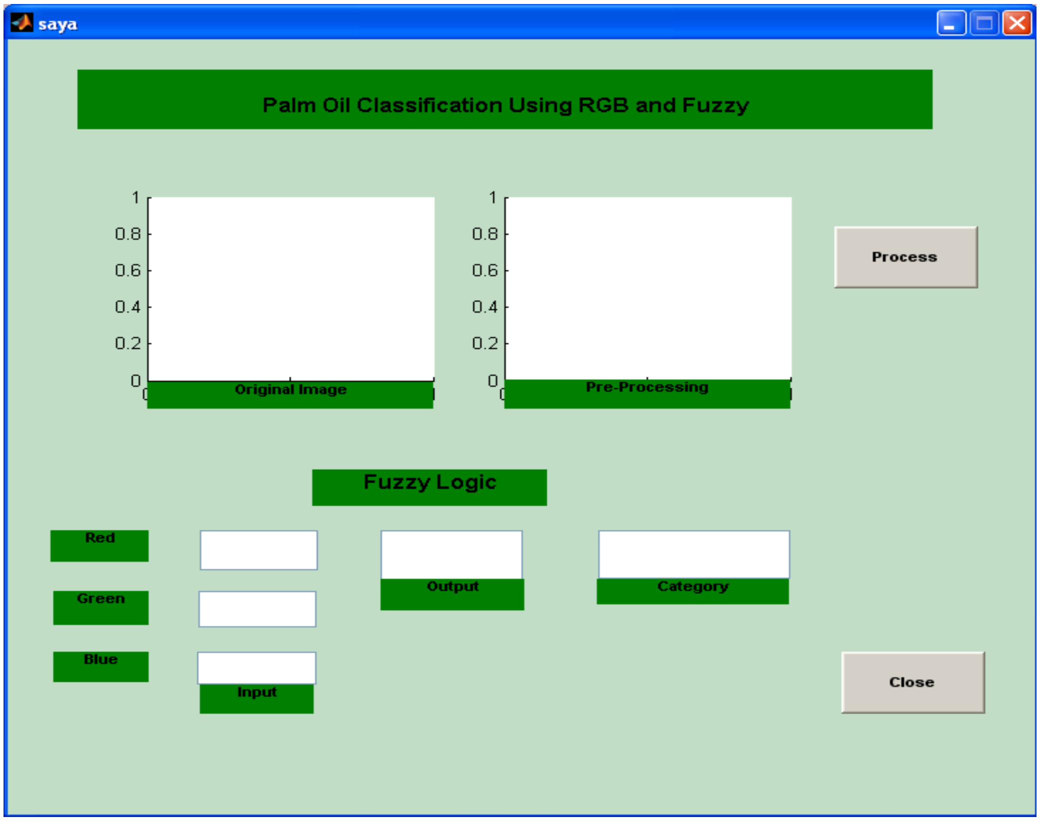
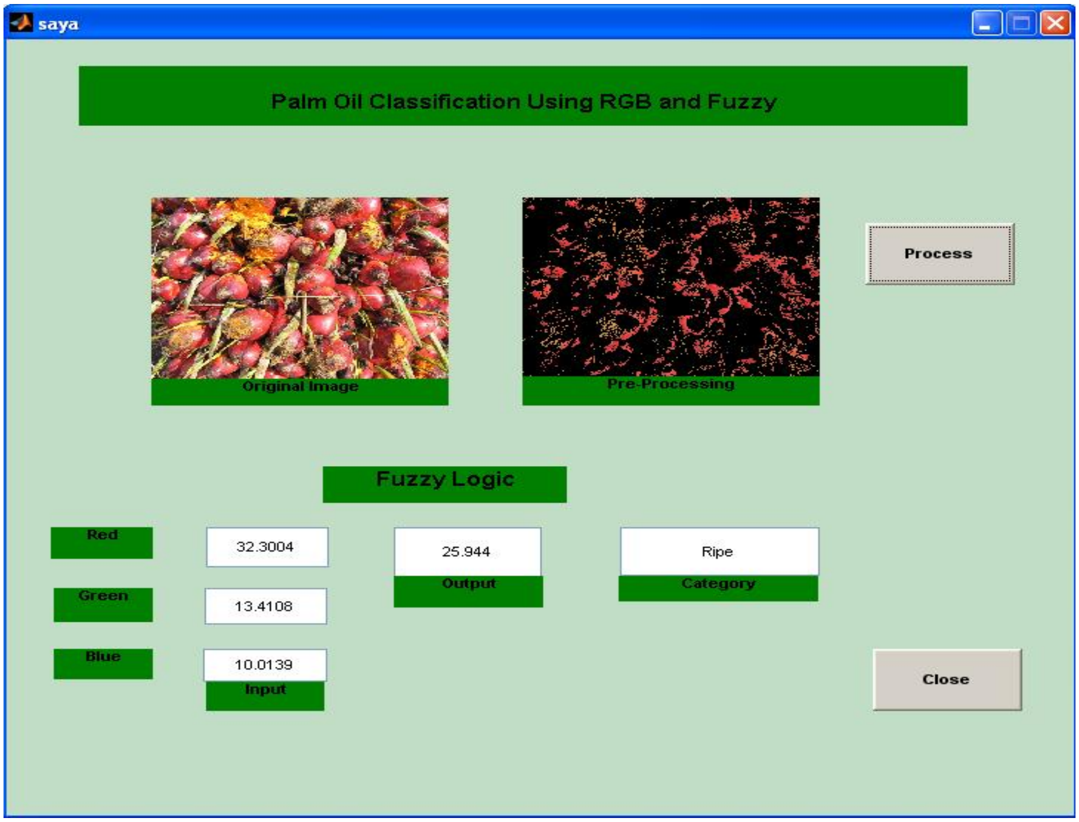


Figure 5.14: Graphical User Interface



Palm Oil Classification Using RGB and Fuzzy

Fuzzy Logic

Red 32.3004

Green 13.4108

Blue 10.0139

Input

Output 25.944

Category Ripe

Figure 5.15: Result display in Graphical User Interface

CHAPTER 6

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Designing and implementation RGB and fuzzy logic technique has been presented in this project. The development software for this project has been done where the fuzzy logic is successfully to give decision in classifying the maturity of oil palm bunch based on the extracted RGB colour feature of the rind.

Oil Palm bunches has been graded by looking at the colour and the content of red component of mesocarp. The ripe category is more reddish mesocarp when compare to others category.

In this project, the main objective are to determine maturity of oil palm fruit bunch and analyze the best structure of fuzzy logic has been successfully. Result and analysis show that human expert can be implement in fuzzy logic, because fuzzy logic can make decision like human thinking. The application of computing technique such as fuzzy logic should be used as classifier in the fruit or agriculture industry.

6.2 Recommendations

This palm oil classification using RGB and fuzzy logic is a prototype or general system that suits applications for other fruit ripeness. It can be implement for improve in agriculture industry.

This project has accomplished the objectives of classification the maturity of oil palm bunches, but still there a lot of idea and invention can be implemented to improve or upgrade this project. It is about 10 images that has been tested for every categories and computed the mean value of RGB. In the future the image can be done more than 10 tested and using more advance camera for capture the image. Besides that the categories of ripeness should be added more such as overripe, empty bunch, small bunches and so on.

The other idea for upgrade this project is using the form of online method within built in or external webcam. For online method, the webcam will capture image in front of it depend on the certain distance. The input will save immediately and pre-processing also runs automatically to display the result at the same time.

REFERENCES

1. Basiron,Y.(2007) Palm Oil Production through sustainable plantations. Europe Journal of Lipid Science Technology, Weinheim : Wiley Interscince
2. Kamarul HG, Rosdiyana S, Nurul WA, Rohana AK. (2009), Image processing Analysis of Oil Palm Fruits for Automatic Grading. Proceedings of the 2009 International Conference on Instrumentation,Control & Automation, October 20-22. Bandung, Indonesia : ISBN 8861-05-5, 75-78.
3. Palm Oil History
   http://en.wikipedia.org/wiki/Palm_oil#History
4. Statistic Oil Palm 2009
   http://econ.mpob.gov.my/economy/Overview_2009.pdf
5. RGB Colour Model
   http://en.wikipedia.org/wiki/RGB_color_model
6. Colour Space
   http://en.wikipedia.org/wiki/Color_space
7. Usmail K. (2003) Apple Grading Using Fuzzy Logic,
   East Lansing MI USA : IEEE, 7864-4/03, 79-81
8. Nursuriati J, Azlinah M, Syazwani A. (2009) Automated Grading of Palm Oil Fresh Fruit Bunches (FFB) using Neuro-Fuzzy. Proceeding of the 2009 International Conference of Soft Computing and Pattern Recognition, 2009, IEEE 3879-2/09, 245-249.
9. M.S.M Alfatni, A.R. Shariff, H.Z. Mohd Shafri, O.M.B. Saaed, and O.M. Eshanta, Oil palm fruit bunch grading using red, green and blue digital number,J. Applied Sciences, vol. 8, no. 8, pp. 1444-1452, 2008.

REFERENCES

10. Timothy J. Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill, 1995.

11. Mathworks
http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/bp78l6_-1.html

12. Mathworks
http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/trimf.html

13. Mathworks
http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/bp78l6_-1.html#bp78l70-5

14. Farah Yasmin A.R, Shah Rizam M.S.B, Ahmad Ihsan M.Y, Nooritawati Md.T and Wan Illia W.I, Monitoring of Watermelon Ripeness Based on Fuzzy Logic, 2009 World Congress on Computer Science and Information Engineering, 2008, IEEE 3507-4/08, 67-70.

15. W. H. Chang et. al., “Vision Based Fruit Sorting System,Using Measures of Fuzziness and Degree of Matching”,Systems, Man, and Cybernetics, 1994. 'Humans, Information and Technology'.,1994 IEEE InternationalConference, 2-5 Oct , San Antonio, TX, USA, vol. 3, pp, 2600- 2604, Oct. 1994.

16. S. Milan et al., Image Processing, Analysis, and Machine,Vision. Brooks/ Cole Publishing Company, 1999.

17. http://www.scantips.com/basics1b.html

18. Manual Penggredan Buah Kelapa Sawit
http://161.142.157.2/pnp/gredffb.htm

REFERENCES

19. M. M. Mokji et al., “Starfruit Grading Based on 2-Dimensional Color Map”, Regional Postgraduate Conference on Engineering and Science (RPCES 2006), Johore, 26-27 July.

APPENDIX A

RESULT

Result Data of Sample Image

Ripe Category	Output Defuzzification	Accepted
1	16.5	Yes
2	25.944	Yes
3	25.2563	Yes
4	16.5	Yes
5	16.5	Yes
6	13.9245	No
7	16.5	Yes
8	24.3991	Yes
9	24.8651	Yes
10	24.2118	Yes

Accuracy = 90%
FRR = 10%
FAR = 0%

Under ripe Category	Output Defuzzification	Accepted
1	14.0435	Yes
2	14.021	Yes
3	14.1115	Yes
4	14.1059	Yes
5	13.9256	Yes
6	4.2938	No
7	4.4812	No
8	9.6506	Yes
9	13.9765	Yes
10	14.0273	Yes

Accuracy = 80%
FRR = 20%
FAR = 0%

Unripe Category	Output Defuzzification	Accepted
1	4.6545	Yes
2	3.2284	Yes
3	3.3219	Yes
4	3.2284	Yes
5	3.2284	Yes
6	3.2284	Yes
7	3.2284	Yes
8	3.2284	Yes
9	3.2284	Yes
10	3.2284	Yes

Accuracy = 100%
FRR = 0%
FAR = 0%

APPENDIX B

SOFTWARE DEVELOPMENT

```
function varargout = saya(varargin)
% SAYA M-file for saya.fig
%      SAYA, by itself, creates a new SAYA or raises the existing
%      singleton*.
%
%      H = SAYA returns the handle to a new SAYA or the handle to
%      the existing singleton*.
%
%      SAYA('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in SAYA.M with the given input arguments.
%
%      SAYA('Property','Value',...) creates a new SAYA or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before saya_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to saya_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help saya

% Last Modified by GUIDE v2.5 26-Nov-2010 01:37:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @saya_OpeningFcn, ...
                   'gui_OutputFcn',  @saya_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before saya is made visible.
function saya_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to saya (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes saya wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = saya_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname]=uigetfile({'*.bmp';'*.jpg';'*.gif';'*.*'},'Pick an Image File');
S =imread([pathname,filename]);
axes(handles.axes1);
imshow(S);
% handles.S=S;
% guidata(hObject,handles);
handles.S=S;
guidata(hObject,handles);

rgbImage=S
% subplot(3, 4, 1);
% imshow(rgbImage);
% title('Original RGB Image');
% Split into color bands.
redBand = rgbImage(:,:, 1);
greenBand = rgbImage(:,:, 2);
blueBand = rgbImage(:,:, 3);
% Display them.
% subplot(3, 4, 2);
% imshow(redBand);
% title('Red Band');
% subplot(3, 4, 3);
% imshow(greenBand);
% title('Green Band');
% subplot(3, 4, 4);
% imshow(blueBand);
% title('Blue Band');

lowRed = 171;
highRed = 255;
lowGreen =40;
highGreen =164;
lowBlue = 30;
highBlue = 92;

% Display them.
% subplot(3, 4, 6);
% imshow(redMask, []);
% title('Red Mask');
% subplot(3, 4, 7);
% imshow(greenMask, []);
% title('Green Mask');
% subplot(3, 4, 8);
% imshow(blueMask, []);
% title('Blue Mask');
% Combine the masks to find where all 3 are "true."
% This will be where the "red" objects are.
redObjectsMask = redMask & greenMask & blueMask;
% subplot(3, 4, 10);
% imshow(redObjectsMask, []);
% title('Combined Mask');
% title('Filled Mask');
% Fill in holes, and display the image.
redObjects = uint8(redObjectsMask);
% subplot(3, 4, 11);
% imshow(redObjects, []);
% title('Filled Mask');
% title('Combined Mask');
% Multiply this mask times the original RGB image
% to get an image that has the red objects only.
maskedImageR = redObjects .* redBand;
maskedImageG = redObjects .* greenBand;
maskedImageB = redObjects .* blueBand;
maskedImage = cat(3, maskedImageR, maskedImageG, maskedImageB);
% Display it.
% subplot(3, 4, 12);
% imshow(maskedImage, []);
% title({'Filled Mask times Original =';'Red Objects Only'});
% figure(4),imshow(maskedImage, []);
% title({'Filled Mask times Original =';'Red Objects Only'});
axes(handles.axes2);
imshow(maskedImage);

redmaskedImage = maskedImage(:,:, 1);
greenmaskedImage = maskedImage(:,:, 2);
bluemaskedImage = maskedImage(:,:, 3);

% %calculate mean r g b
 Xr = mean(mean(redmaskedImage))
  set(handles.edit4, 'String', Xr)
 Xg = mean2(greenmaskedImage)
  set(handles.edit5, 'String', Xg)
 Xb = mean2(bluemaskedImage)
  set(handles.edit6, 'String', Xb)
%   get(hObject,'String')
% decision making
%     if 0<Xr<10
% %         disp ('unripe')
%     set(handles.edit3, 'String', 'unripe');
%     elseif 7.3<=Xg>=20
% %         disp ('Under ripe')
%     set(handles.edit3, 'String', 'Under ripe');
%     elseif 15<Xb>33
% %         disp ('Ripe')
%     set(handles.edit3, 'String', ' Ripe');
%     end
 % guidata(hObject,handles)
handles.maskedImage= maskedImage;
guidata(hObject,handles)

p = str2double(get(handles.edit4,'String'));
q = str2double(get(handles.edit5,'String'));
r = str2double(get(handles.edit6,'String'));
a=newfis('trapntrim');
```

```
a = addvar(a,'input','w',[0 33]);
a = addmf(a,'input',1,'Small','trapmf',[-11.9 -1.32 1.18 3.01]);
a = addmf(a,'input',1,'Medium','trimf',[1.79 6.24206349206349 11.2]);
a = addmf(a,'input',1,'Large','trapmf',[14.0119047619048 26 34.3 44.9]);


a = addvar(a,'input','x',[0 33]);
a = addmf(a,'input',2,'Small','trapmf',[-11.7 -1.14 1.62 7.11507936507937]);
a = addmf(a,'input',2,'Medium','trimf',[3.88 11.8 15.1468253968254]);
a = addmf(a,'input',2,'Large','trapmf',[16.9801587301587 22.8269841269841
33.4269841269841 44.0269841269841]);

a = addvar(a,'input','y',[0 33]);
a = addmf(a,'input',3,'Small','trapmf',[-10.6 -0.0119 1.18 4.67063492063492]);
a = addmf(a,'input',3,'Medium','trimf',[7.11507936507937 10.6 13]);
a = addmf(a,'input',3,'Large','trapmf',[15.0595238095238 22.7 34.4 45]);

a = addvar(a,'output','z',[0 33]);
a = addmf(a,'output',1,'Unripe','trapmf',[-11.9 -1.32 1.32 9.82142857142857]);
a = addmf(a,'output',1,'Under_ripe','trimf',[7.28968253968254 15.2 20]);
a = addmf(a,'output',1,'ripe','trapmf',[14.797619047619 31.7 34.3 44.9]);

rulelist=[...
3 1 1 3 1  1
3 2 1 3 1  1
3 2 2 3 1  1
2 1 1 2 1  1
2 2 1 2 1  1
1 1 1 1 1  1
1 2 2 1 1  1];

a=addrule(a,rulelist);

z = evalfis([p q r], a)

set(handles.edit2,'String',num2str(z))
set(handles.edit2,'Value',z)


function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close();

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```