

**AUTOMATED FRUIT AND FLOWER COUNTING USING DIGITAL
IMAGE ANALYSIS**

HOO ZHOU YANG

**A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Bachelor of Engineering (Hons) Electronic Engineering**

**Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman**

September 2015

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : Hoo Zhou Yang

ID No. : 10AGB05052

Date : _____

APPROVAL FOR SUBMISSION

I certify that this project report entitled “**AUTOMATED FRUIT AND FLOWER COUNTING USING DIGITAL IMAGE ANALYSIS**” was prepared by **HOO ZHOU YANG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons) Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor: Dr. Humaira Nisar

Date : _____

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2015, Hoo Zhou Yang. All right reserved.

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Humaira Nisar for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents, girlfriend and friends who had helped and encouraged me.

AUTOMATED FRUIT AND FLOWER COUNTING USING DIGITAL IMAGE ANALYSIS

ABSTRACT

The purpose of this project is to predict the yield of fruit and flowers. The ability to predict the yield would benefit the farmers as they plan the sale, the shipment and operations. In this project we have used digital images to segment the fruit and flowers. The proposed algorithm includes image segmentation, size thresholding and shape analysis, counting of the regions of interest, and yield prediction. We have used two colour spaces RGB and YCbCr. The percentage error quantification for RGB model(R-G) is 8.75% for dragon fruit and 11.30% for daisy while for YCbCr model(C) percentage error is 8.07% for dragon fruit and 5.54% for daisy. Based on our analysis we have observed that YCbCr gives better results. Finally result of regression analysis for dragon fruit and daisy are 0.9517 and 0.9751 respectively. The average percentage error in yield prediction for dragon fruit is 1.40% and daisy is 5.52%.

TABLE OF CONTENTS

DECLARATION	ii
APPROVAL FOR SUBMISSION	iii
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF SYMBOLS / ABBREVIATIONS	xiii
LIST OF APPENDICES	xiv

CHAPTER

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statement	2
	1.3 Aims and Objectives	3
2	LITERATURE REVIEW	4
	2.1 Image acquisition	4
	2.2 Image Noise Removal	5
	2.3 Image Segmentation	5
	2.3.1 Thresholding segmentation technique	5
	2.3.2 Improved Watershed transform segmentation	6
	2.3.3 R-G chromatic aberration characteristic dynamic threshold segmentation method.	7

	2.3.4	LVQ network adaptive segmentation method.	8
	2.3.5	Conditional Random Fields (CRF) algorithm	8
	2.3.6	TsNKM method	9
	2.4	Object Recognition	9
3		METHODOLOGY	19
	3.1	Project Management	19
	3.2	Image Acquisition Method	20
	3.3	Filter Selection for Noise Removal	20
	3.4	Concept of Image Segmentation	21
	3.5	Concept of shape analysis	23
	3.6	Automated Fruit and Flower Counting	25
4		RESULTS AND DISCUSSIONS	26
	4.1	Colour Space Selection	26
	4.2	Result for Dragon Fruit	32
	4.2.1	Automated Dragon Fruit Counting result for RGB and YCbCr colour space	32
	4.2.2	Automated Dragon Fruit Counting result for RGB and YCbCr colour space and shape analysis.	38
	4.2.3	Summary of Automated Dragon Fruit Counting	43
	4.3	Result for Daisy	44
	4.3.1	Daisy Segmentation and Automated Counting result for RGB(R-B) colour space	44
	4.3.2	Daisy Segmentation and Automated Counting result for YCbCr(Cr) colour space	48
	4.3.3	Summary of Daisy Automated Counting	53
	4.4	Limitation of fruit and flower automated counting	54
	4.5	Yield Prediction	54
	4.5.1	Dragon Fruit Yield Prediction	54
	4.5.2	Daisy Yield Prediction	56
5		CONCLUSION AND RECOMMENDATIONS	59

		ix
5.1	Conclusion	59
5.2	Recommendations	60
REFERENCES		61
APPENDICES		64

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1:	Comparison of the advantages and disadvantages of different techniques.	13
Table 4.1:	The input images and segmented images in RGB and YCbCr colour space.	27
Table 4.2:	Comparison of counting result for dragon fruit (colour space) to manual counting.	32
Table 4.3:	Comparison of counting result for dragon fruit (colour space + shape analysis) to manual counting.	38
Table 4.4:	The summary of the results obtained from different colour space and shape analysis.	43
Table 4.5:	Comparison of counting result for daisy flower (colour space) to manual counting.	44
Table 4.6:	The result obtained by the Automated counting (Cr) and compared to the manual counting.	49
Table 4.7:	The automated counting result and regression analysis result obtained from RGB and YCbCr colour space	54
Table 4.8:	The predicted dragon fruits yield	55
Table 4.9:	The input images and automated counting result.	55
Table 4.10:	The percentage error of the predicted yield for daisy	57
Table 4.11:	The predicted yield of Daisy Flower	57

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1:	Original image	7
Figure 2.2:	R-G gray level image	7
Figure 3.1:	Gantt chart for final year project 1	19
Figure 3.2:	Gantt chart for final year project 2	20
Figure 3.3:	The comparison between input image and Gaussian filtered image	21
Figure 3.4:	Convert grayscale image into binary image by thresholding of 0.6.	22
Figure 3.5:	Holes filled up in the segmented region.	22
Figure 3.6:	Image erosion of segmented region.	23
Figure 3.7:	The roundness value obtained from Matlab	24
Figure 3.8:	The flow chart of the algorithm with colour segmentation and shape analysis.	25
Figure 4.1:	The colour Space threholding comparison(Red channel, R-G channel and C_r channel)	27
Figure 4.2:	Regression analysis between the no. of fruits counted by RGB automated and manual counting.	37
Figure 4.3:	Regression analysis between the no. of fruits counted by YCbCr automated and manual counting.	37
Figure 4.4:	Regression analysis between the no. of fruits counted by RGB combined shape analysis automated and manual counting.	42

- Figure 4.5: Regression analysis between the no. of fruits counted by YCbCr combined shape analysis automated and manual counting. 43
- Figure 4.6: Regression analysis between the no. of flowers counted by RGB automated and manual counting. 48
- Figure 4.7: Regression analysis between the no. of flowers counted by YCbCr automated and manual counting 53

LIST OF SYMBOLS / ABBREVIATIONS

A	area of the segmented region
P	perimeter of the segmented region
R	roundness of the segmented region
R^2	Total variation
RGB	Additive colour model (Red, Green and Blue)
YCbCr	Luminance, Chrominance Blue and Chrominance Red

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
APPENDIX A:	Computer Programme Listing	64

CHAPTER 1

INTRODUCTION

1.1 Background

Recently image processing techniques have been proved to be important tool for analysis in different fields and applications. Image processing in agriculture has been applied in the areas of sorting, grading, detection and counting the products. Precision agriculture was developing technology to enhance farm output and help farmers get greater control over the management of operations.

In agriculture field the counting of the number of fruits and flowers play an important role to estimate the amount of harvest. The counting total number of fruits and flowers in a farm using human power is a very tiring job and need plenty of time to complete the task. The image processing techniques can help to perform accurate fruit counting. Thus, automated fruit and flower counting was introduced in the agriculture field by using digital still images to count the total number of fruits or harvest in that image. The automated counting technique can predict the yield of the farm. The automated fruit and flower counting consists of five steps which are image acquisition, image noise removal, image segmentation, object recognition, automated fruit and flower counting and lastly the yield prediction as shown in Figure 1.1.

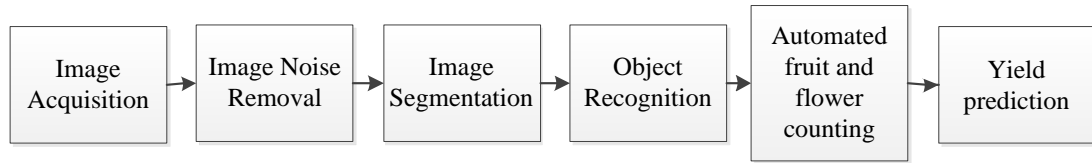


Figure 1.1: Overview block diagram of the project.

1.2 Problem Statement

Manual counting of products in farm may lead to bad estimations due to the inaccuracy with manual counting. If overestimated, it will cause the farm to lose money on the shipping part since ordered more placements and the harvested product is lesser. In the other hand, the products were underestimated; the farm will suffer from insufficient pickers and packer staff to handle the bigger amount of harvested product. Thus, the pre-order shipment will need to add-on extra weight to ship all fruit out to desired destination. There is no automated system designed to count the total number of fruits or flowers before harvesting. Thus, the automated fruit and flower counting technique will be a very helpful system for the farm.

This project is mainly based on digital image analysis and programming is done using MATLAB. Image processing includes the image analysis and image segmentation. The image analysis in this project is the process of extracting useful and important information from image such as shape, color, and counting objects. Image segmentation refers to the partition of an image into a set of regions that cover it. The goal of image segmentation is to represent meaningful areas which mean the fruit and flower in this project.

1.3 Aims and Objectives

The objectives of the project are as follows:

- i) To segment fruit and flower (dragon fruit and daisy) from the images by using image processing techniques.
- ii) To quantify number of dragon fruit and daisy from images.
- iii) To predict yield of fruit/flower for agricultural automation.

CHAPTER 2

LITERATURE REVIEW

2.1 Image acquisition

The first step of the automated fruit counting is input image acquisition. The light source, the picture background and the distance between camera and object requires to be controlled to improve the quality of the digital image and image segmentation of interested object (Dorj, Lee and Lee, 2013).

The image with different illumination conditions is more difficult to identify the fruit due to the variation in color intensities in images. A process of removing effect of illumination conditions was applied before identifying the fruit known as normalization of illumination condition. Normalization of illumination condition minimizes the drastic change in intensity level within an image. The equation of normalization of illumination condition is shown below.

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = 255 \begin{pmatrix} R/I \\ G/I \\ B/I \end{pmatrix} \quad (2.1)$$

$$\text{where, } I = 0.2989R + 0.5870G + 0.1140B \quad (2.2)$$

After the normalization, the illumination level throughout the image becomes approximately uniform so the colour variation is reduced due to the different illumination conditions (Choi, Lee and Reza, 2013).

2.2 Image Noise Removal

First, the Gaussian Filter is used to reduce the background noise of the image in great scale. According to Dorj, Lee and Lee (2013) research the Gaussian 2x2 filter is the most optimal filter among the Gaussian 1x1, 2x2 and 3x3.

The input image is pre-processed to eliminate poor illumination conditions at the corners which only consider the center window size of 825x540. The intensity of the image was transformed non-linearly to keep brighter pixel less affected and contrast darker pixels in a wider range (Wifethunga, et al., 2008). The RGB components of the image also rearranged to make the image false colour and the kiwifruit will easily distinguished from the background of the input image. A window size 3x3 pixels window sized average filter was used to filter the image noise (Wifethunga, et al., 2008)

2.3 Image Segmentation

2.3.1 Thresholding segmentation technique

Thresholding technique is widely used for image segmentation because of its simplicity and able to separate the selected image into numerous areas based on the gray levels of the image. Manual threshold selection is done by trial and error by using the histogram of the selected image (Sarkate, Kalyankar and Khanale, 2013). Clustering technique separates different areas based on the similarity without prior information. There are two clustering techniques which are fuzzy method and crisp method. Fuzzy method is suitable for more than one cluster while the crisp method is suitable to classify only one cluster. FCM (Fuzzy C-means) method is able to classify data into numerous clusters accurately and automatically. FCM is sensitive to the variation of illumination thus it is effective to segment images under controlled illumination conditions.

The colour is the fundamental feature of digital image. An algorithm can be based on colour to differentiate the interested objects from the image (Hung, et al., 2013). L*a*b colour space can be used for segmentation of a fruit image (Anisha, Divya and Shanu, 2013). The RGB colour value can be used to segment fruit from the input image at specific R-B colour value threshold based on the colour intensity of the interested fruit. Apples in the image can be segmented from background with the threshold colour differences of R-B by analysing the colour values for each colour class of RGB. (Anisha, Divya and Shanu, 2013).

The Hue, saturation and value in the HSV colour space are used in algorithm as visual cues for the red apple detection. Thus, red apples can be distinguished by colours under controlled illumination. The hue value of red apple pixel is in the range of 0° to 9° and from 349° to 360° . Firstly, segment the pixels with hue value within 0° to 9° or 349° to 360° and remove the image background pixels with saturation value of 0.1 or smaller than 0.1. Then, the regions of red apple are segmented from the input image (Wang, et al., 2012). Two major problems caused by large clusters of apples which are some apples cannot be detected due to the occlusion by other apples nearby and current version of software algorithm developed only able to split a cluster into two apples (Wang, et al., 2012).

2.3.2 Improved Watershed transform segmentation

Over-segmentation is a serious problem for most of the watershed algorithms which classify as a region based segmentation technique. Image enhancement techniques are used to avoid under segmentation and noise removal to reduce over segmentation. The watershed transform able to separate touching objects in an image by finding 'catchment basins' and watershed ridge lines' by treating it as a surface where light pixels are high and dark pixels are low. The initial watershed transform of gradient images result the homogeneous regions fragmented into small region. Thus, neighbourhood relations introduced to merge the fragmented region. The watershed line corresponds to the most significant edges between the markers and not affected

by lower-contrast edges. The watershed transform can detect a contour even there are no strong edge between markers (Bala, 2012).

2.3.3 R-G chromatic aberration characteristic dynamic threshold segmentation method.

Figure 2.1 and 2.2 shows the dynamic threshold segmentation base on R-G chromatic aberration characteristic of tomato image. There is a big difference between tomato and background from figure 2.1. R-G (Red channel minus Green channel) gray value of tomato is larger than the background.



Figure 2.1: Original image

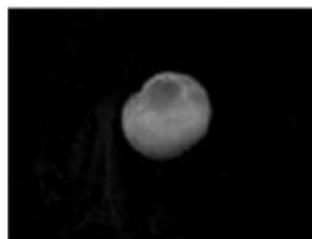


Figure 2.2: R-G gray level image

For R-G gray-level image with size of M rows and N columns $f(i,j)$, the segmentation based on equation 2.3 and obtained the BW binary image, as in

$$BW = \begin{cases} 255, & f(i,j) \geq T \\ 0, & f(i,j) < T \end{cases} \quad (2.3)$$

Global threshold T by Otsu method, is given by

$$T = \text{Average}[\max_{0 < t < m-1} \sigma_B^2(t)] \quad (2.4)$$

where $\sigma_B^2(t)$ is the group variance between fruit class and background class, m is maximal gray value.

Based on the global threshold above, the fruit pixels are set to white color (255) and background pixels set to black (0).

2.3.4 LVQ network adaptive segmentation method.

Learning Vector Quantization (LVQ) is a supervised version of vector quantization with labelled input data. This learning method uses training data made up of color coordinate values of fruit and backgrounds (Yin, Mao and Xie, 2008). The LVQ has binary output to represent the fruit and background.

The classes are predefined as fruit image segmentation by trained LVQ by colour coordinate values of tested fruit image fed into LVQ model and binary image is obtained after computation of the model. Then, fruit region pixels are set to white colour and background pixels are set to black colour. The disadvantages of LVQ are poor segmentation, high cost of time and cannot apply in real-time (Yin, Mao and Xie, 2008).

2.3.5 Conditional Random Fields (CRF) algorithm

The Conditional Random Fields (CRF) algorithm is able to segment different objects and also the background scene as the sky and ground from the image; consists of leaves, fruit and tree trunks at different lighting condition and scales. The majority of misclassification occurs between the green apple and leaf classes due to the similar colour and texture characteristic (Hung, et al., 2013). The image segmentation

required to consider all the specification of the fruit to get an accurate segmentation of interested object such as shape, colour, intensity and surface texture.

2.3.6 TsNKM method

The method namely TsNKM integrates two algorithms, which are improved thresholding and adaptive K-means. This method uses LAB format image, the improved thresholding produce good segmented image for black objects while adaptive K-means handle segmentation of object with other colors (Hamirul, et al., 2014).

2.4 Object Recognition

Recognition system is an important field of image processing which is involved in the visual recognizing patterns of the interested objects. The recognition and classification of digital still 2D fruit images are colour based and shape based analysis methods. The system combines all features such as colour, shape, size and other characteristics of the fruit for recognition to increase the accuracy.

The K-Nearest Neighbors(KNN) algorithm is used for image classification, the unknown object will be compared to the data set being saved in the classification algorithm(woo and Mirisae, 2009). The distance between the features values of the data set and the input fruit image were determined by the Euclidean distance (woo and Mirisae, 2009). The KNN algorithm is an effective classification algorithm to be used in fruit recognition system with the accuracy up to 90%.

The fruit recognition methods used in Song, Glasbey, Horgan, Polder, Dieleman and Heijden (2013) research are initial points of interest and bag-of-words (BoW) model. The points of interest include colour transformations and colour classifier. The RGB colour intensity was transformed to differentiate the fruit and

other plant parts, these transformations are done which are G-B, G-R and the final transformation $G/(R+G+B)$. This colour transformation is more sensitive compared to original R, G and B values than changing the illumination conditions (Song, et al., 2013). After the colour transformation, the colour classifier was applied separately to every pixel in the image and the probabilities of fruits were combined. The Bag of Words model combined with the use of Maximally Stable Colour Region (MSCR) to detect the highly variable fruit shape. A fruit is determined in a support window with two different feature sets which are maximally stable colour region features and texture features obtained. The colour difference between the largest and smallest intensity in the filter window calculated by a local range filter.

The system will recognize the grape based on colour mapping, morphological dilation and black area. The morphological dilation operation performed to fill up the gaps between pixels and uniform black regions. A square of 3x3 pixels used as the structuring element, typically 60 iterations of morphological dilation for red grapes while 100 iterations for white grapes. When the size of the structuring element increases, the number of overlapping uniform black regions increases as well even with a lower number of morphological dilation iterations (Reis et al., 2012). Thus, the overlaps will result in incorrect identifications. The system is able to detect 91% of white grapes and 97% of the red grapes of correct classification. The extra function of this system is that it can calculate the location of bunch stem and help the guiding system of robots (Reis, et al., 2012).

2.5 Automated Fruit Counting

In general, there are many related research and studies conducted in Malaysia and foreign countries to develop various fruit detection methods for automated fruit counting. The automated fruit counting is very useful in agriculture and plantation fields but the uncertainty of agricultural objects are difficult to employ the existing algorithm to the agricultural field.

There are few approaches are used to estimate the fruit count after the segmentation output. The pixel count used to deduce the actual fruit count from the total number of fruit pixels per side of the row. Thus, the total number of fruit can be counted. Before the detection of fruit, the noise is removed by image erosion and the occluded fruit were joined partially by dilation. The second approach used the circular Hough transform to estimate the actual fruit count by perform circle detection (Hung, et al., 2013). The algorithm was re-designed to assess the fruit counting performance with specialize in apple and non-apple binary classification. The result obtained was the classification accuracy improved with the decreasing of the number of classes (Hung, et al., 2013).

A good segmentation result provided by L*a*b color space image with K-means clustering or color based nearest neighbor classifier. The circle fitting algorithms applied after the morphological operation to partition all connected fruit in the image right after the color based segmentation (Anisha, Divya and Shanu, 2013). Thus, the counting of fruit can be completed easily with the algorithms applied. The fruit counting based on the fuzzy logic classification for fruit recognition and counting are efficient (Anisha, Divya and Shanu, 2013).

The input image required to convert into gray color then perform regional maximum (RM), Distance Transform (DT) and area calculation for automated counting (Wijethunga, et al., 2008). The center area of the fruit segment has the higher gray color values, the total number fruit count obtained by the number of the regional maximums of the input image. Next, the distance transform is performed but it requires the input image as binary form. After the converting into binary form, the fruit pixels in segmentation have the value of 0 and the non-interested pixels have the value of 1. The nearest non-zero valued pixel between each pixel in the fruit region was calculated. The total area of the fruit in the image was calculated. The relationship between the true count and database was analysed using linear regression with the help of Matlab programming. Based on the observation, the area method is suitable to check whether there are no fruit in the image (Wijethunga, et al., 2008). The errors in counting technique cause by the occluded fruit in the image. The fruit count will be underestimated when many fruit connected as one object with the distance transform method. While the regional maximum method will

overestimate the fruit count due to too many local maximum. The error can be neutralized when both of the overestimated and underestimated occurred in one image (Wijethunga, et al., 2008).

The yield prediction use binary mask image and shape analysis. First, region labelling based on the 4-connectivity by scans the whole image raw wise and the edge detection by applied the Sobel edge detector for labelled region. A 3x3 Sobel operator used for the edge detection in horizontal and vertical direction as below

$$\begin{aligned}
 Sx &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \\
 Sy &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
 \end{aligned}
 \tag{2.5}$$

Lastly, find the coordinates of edge pixels and the mean of pixels. Next, the center of the cluster found by using the mean calculated and the approximated circle was fitted into the data set. The probable fitted circle plotted on the input image (Patel, Jain and Joshi, 2012). The accuracy of the algorithm is 98% based on the difference between manual counting and fruit count from the algorithm.

Table 2.1: Comparison of the advantages and disadvantages of different techniques.

Author	Year	Object type	Technique used	Advantages/ Disadvantages	Image used	Accuracy
C. Hung, J. Underwood, J. Nieto, and S. Sukkarieh	2013	Apple	<ul style="list-style-type: none"> -Conditional Random Fields -Remove noise using image erosion -Partially blocked fruit joined by image dilation -Circle detection using circular Hough transform 	<p>Advantages</p> <ul style="list-style-type: none"> -The apple count can be estimated by the pixel classification output <p>Disadvantages</p> <ul style="list-style-type: none"> -The algorithm runs at 30seconds per frame. 	8000	93.3%
Dorj, U.O., Lee, K.K. Lee, M	2013	Tangerine	<ul style="list-style-type: none"> -Gaussian 2x2 filter to remove noise -L*a*b color space -RGB color detection algorithm -Sobel Edge detection 	<p>Disadvantages</p> <ul style="list-style-type: none"> -Variance in natural illumination makes pixel data based method difficult to tune and work reliably. 	172	73%

Q. Wang., Stephen, N., Marcel, B., Sanjiv. S.	2012	Apple	<ul style="list-style-type: none"> -Color segmentation using hue, saturation and value in HSV color space -Hue and saturation thresholding -morphological method -Detect apple regions with specular reflections 	<p>Disadvantages</p> <ul style="list-style-type: none"> -Some apples are invisible and cannot be detected due to the occlusion. -The algorithm only can split a cluster into two apples. 	NA	96.8%
Y. Song., C.A. Glasbey,, G.W. Horgan., G. Polder,, J.A. Dieleman., G.W.A.M. van der Heijden	2013	Green& red pepper	<ul style="list-style-type: none"> -Initial points of interest -Color classifier -Bag-of-Words Model 	<p>Disadvantages</p> <ul style="list-style-type: none"> -Many false positives using the initial color classifier 	NA	74.2%

Wijethunga, P., Samarasinghe, S., Kulasiri, D., Woodhead, I.	2008	Gold& Green kiwifruit	<ul style="list-style-type: none"> -Use $L^*a^*b^*$ color space -Morphological open to remove noise -Use Regional Maximum(RM), Distance Transform(DT) and area for counting. 	<p>Disadvantages</p> <ul style="list-style-type: none"> -Poor segmentation for green kiwifruit due to misclassifies similar colored part as kiwifruit. -Overestimation for Regional maximum method, underestimation for distance transform. -Area counting method based on the size of interested object depend on the distance to camera. 	NA	Gold 89.72% Green 68.72%
Woo, C.S., Mirisae, S.H.,	2009	14 type of fruits	<ul style="list-style-type: none"> -RGB color model -K-Nearest neighbors algorithm -recognize fruit by using color values, shape roundness values, area and perimeter values -50 fruits images for database 	<p>Advantages</p> <ul style="list-style-type: none"> -Color based, shape based, size based methods combined increase the accuracy 	NA	90%

M.J.C.S. Reis, R. Morais, E. Peres, C. Pereira, O. Contente, S. Soares, A. Valente, J. Baptista, P.J.S.G. Ferreira f, J. Bulas Cruz	2012	White & Red Grape	-Morphological dilation - structuring element of 3x3 pixels	Disadvantages -The algorithm does not able to count the total number of grape in the image.	225	White grape: 91% Red grape: 97%
Anisha, S., Divya, G., Shanu, S	2013	Apple	-Gaussian low pass filter to remove noise -RGB threshold segmentation -L*a*b color space used for segment apple. -Fuzzy classifier and Fuzzy logic used for extract features of apple.	Not Mentioned	NA	NA

D, Choi., W. S. Lee., Reza, E	2013	Citrus	-L*a*b color space -Morphological filling holes operation -Entropy texture analysis	Disadvantages -The algorithm treat the highly saturated soil and leaf had bright yellowish color as the citrus.	622	81.3%
Hamirul A.H., Sharifah L. S. A., Nursuriati J., Hazaruddin H	2014	Jatropha	-L*a*b color space -Improved thresholding and adaptive K-means -Similarity index(SI) to match the fruit	Advantages -The adaptive K-means algorithm able to segment images accurately. -The improved thresholding algorithm capable to segment black images perfectly. Disadvantages -TsNKM hardly to segment non-circular shaped object	120	NA
Sarkate, R.S., Kalyankar, N.V., Khanale, P.B.,	2013	Flower	-HSV color space -thresholding technique	Disadvantages -The algorithm unable to separate overlapping flowers	75	86.58%

Bala. A	2012	Pears	-Improved watershed transform -Prewitt's operator	Advantages -Watershed transform not affected by lower contrast edges -Able to detect contour with weak edges between markers.	NA	NA
Yin. J., Mao. H., Xie. Y	2008	Apple, Oranges, Tomato	-RGB color space -Dynamic threshold segmentation -Adaptive LVQ network segmentation	Advantages -Dynamic threshold segmentation use less time compared to other algorithm used in this paper. Disadvantages -High cost time and poor performance for adaptive LVQ network segmentation method.	754	NA
Patel H. N., Jain R.K., Joshi M. V	2012	Orange	-L*a*b color space -Gaussian filter -Sobel operator -Morphological operation -Circle fitting algorithm	Advantages -Able to calculate yield very accurately in clustered background.	100	98%

CHAPTER 3

METHODOLOGY

3.1 Project Management

In final year project 1, a project title proposal was submitted to the respective supervisor and the approval for the project title was made in first two week in the trimester. The literature review started in week 2 and the basic image processing and coding of Matlab started from week 3. The design of the algorithm started in week 6. The reporting of project started in week 8 and the final year project 1 report was submitted in week 12, the oral presentation in week 14 as shown in the Figure 3.1.

Details/week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Project Proposal	■	■												
Literature Review		■	■	■	■									
Study basic of image processing			■	■	■	■								
Study the coding of Matlab			■	■	■	■	■							
Design software						■	■	■	■	■	■	■	■	■
Reporting							■	■	■	■	■	■	■	■
Oral Presentation														■

Figure 3.1: Gantt chart for final year project 1

In final year project 2, the project work is continued from the progress of final year project 1. The improvement of the object segmentation has been done. The final report is submitted in week12 and final oral presentation is in week 14 as shown in the figure 3.2.

Details/week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Project work(continue)														
Reporting														
Final Report submission														
Oral Presentation														

Figure 3.2: Gantt chart for final year project 2

3.2 Image Acquisition Method

The image acquisition is the first step of this project and the quality of the image plays an important role in the coming image processing steps. First, the distance between camera and fruit tree should be considered so the fruit size in the image obtained will not have a huge variation in size. Besides that, all the images should be taken under controlled illumination condition. All images were taken from the internet database for this project to run the image processing analysis. However, the images obtained from the internet had different distances between camera and the fruit tree and different illumination condition of the images. Hence, a good algorithm or method needed to solve this problem.

3.3 Filter Selection for Noise Removal

The smoothing filter is used for noise reduction and blurring effect in the image. The smoothing filter applied to remove small details and bridging small gaps in contour. The output of the smoothing filter was the average of the surrounded neighbourhood

pixels of the filter mask, the smoothing effect depend on the filter mask size. The smoothing filter replaces the value of every pixel in the image by the average of the intensity levels in the neighbourhood defined by the filter mask. The random noises from the image typically consists of sharp transitions in the intensity levels, thus the smoothing filter is able to perform the noise reduction effectively. After trial and error in Matlab, Gaussian filter size of 3x3 had the best performance in remove noise of the input image and the edge of edge of interested object still can be seen as shown in figure 3.3. Thus, the 3x3 Gaussian filters applied to remove noise before processed to the next step.



Figure 3.3: The comparison between input image and Gaussian filtered image

3.4 Concept of Image Segmentation

Image segmentation is the process of partitioning the image into numerous segments to locate the object of interest. There are several techniques and algorithms to perform the segmentation in the digital images such as thresholding, clustering, edge detection, histogram-based methods etc. In this project, the thresholding method is used to segment the fruit.

The input image was converted into YCbCr colour space and shown in Red chrominance. Next, the red chrominance grayscale image converted into binary image with threshold of 0.6 with the pixels of fruit in white colour and the pixels of the background is black colour, as shown in figure 3.4.



Figure 3.4: Convert grayscale image into binary image by thresholding of 0.6.

The segmented region will have some holes or noise inside the region in the binary image of figure 3.4, the “imfill” function was used to fill up holes in the binary segmented image. The holes and noise inside the segmented region has filled up with white pixels as shown in the figure 3.5.

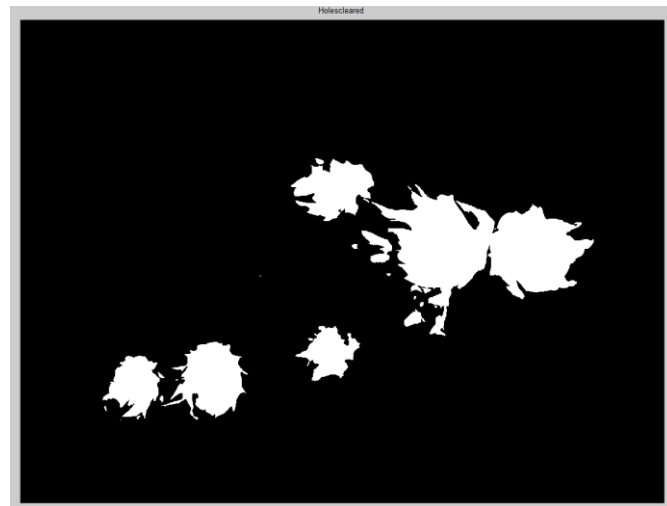


Figure 3.5: Holes filled up in the segmented region.

Next, the image erosion was applied to the binary image in figure 3.5 to separate the clustering fruit region into 2 separate regions as shown in figure 3.6. The image erosion uses the diamond mask with the size of 20.

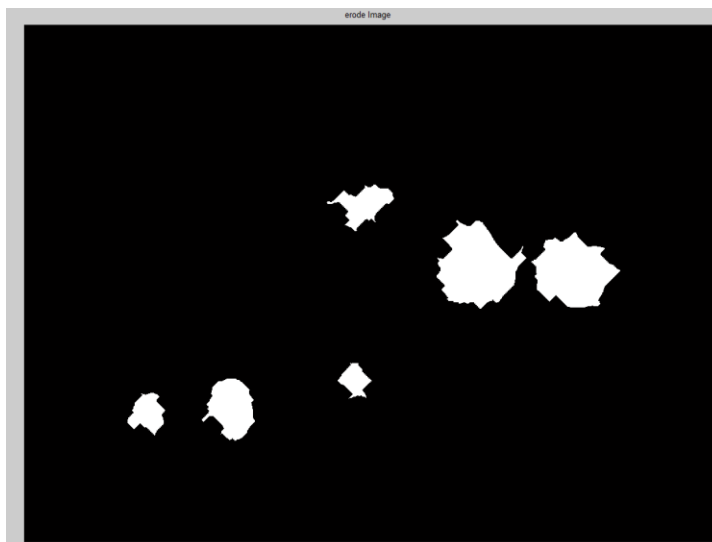


Figure 3.6: Image erosion of segmented region.

Lastly, the morphological opening function applied on the binary segmented image followed by the morphological closing to further remove noises in the segmented image.

3.5 Concept of shape analysis

Shape analysis is to analyse and process the geometric shape in the segmented region of interest; in this project the roundness analysis is used to increase the accuracy of the object detection. The area and perimeter of each segmented regions will be estimated. The boundary coordinate is used to estimate the area and perimeter of the segmented regions using morphological function with the extracted boundary.

The roundness value of the segmented regions are obtained by applying the formula

$$R = \frac{4\pi A}{P^2} \quad (3.1)$$

where

R= roundness of the segmented region

A= area of the segmented region

P= perimeter of the segmented region

The roundness value is equal to 1 for a circle and its value less than 1 for any other shape. From trial and error, the roundness value between 0.45 and 0.60 mostly are the occluded or overlapped interested objects. For the segmented regions having the roundness values from 0.45 to 0.60 will be counted as two fruit as shown in figure 3.7.



Figure 3.7: The roundness value obtained from Matlab

3.6 Automated Fruit and Flower Counting

The automated fruit and flower counting use the size threshold for the segmented region. Once the segmented region area greater than a specified threshold area in the total area in the image. Any white area greater than the threshold area will be counted as 1 dragon fruit found in the image. Next, the shape analysis will count the segmented regions with roundness value between 0.45 and 0.60 as one more dragon fruit. The total fruit count is the sum of the fruit count in size threshold and shape analysis. The process flow of the automated fruit and flower counting is shown in Figure 3.8.

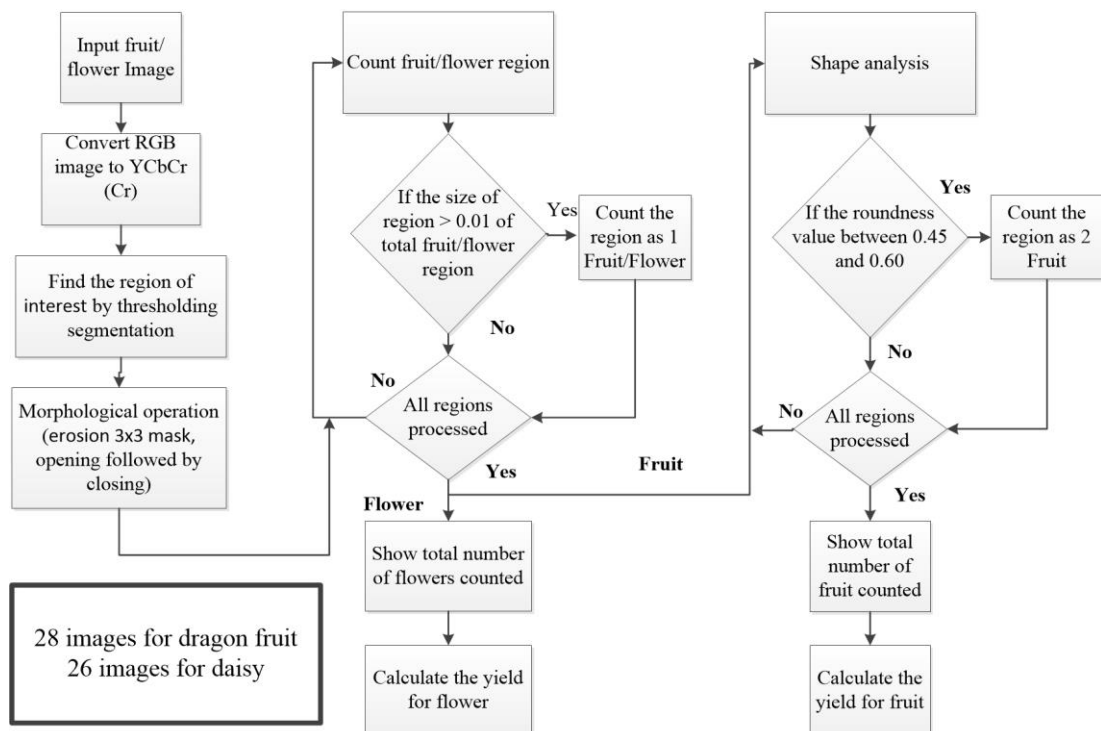


Figure 3.8: The flow chart of the algorithm with colour segmentation and shape analysis.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Colour Space Selection

The colour space used in image segmentation is YCbCr and RGB colour space. YCbCr consists of Luminance (Y), Chrominance Blue (C_b) and Chrominance Red (C_r). The YCbCr colour spaces are chosen for the image segmentation as the C_r is strong in places of occurrence of reddish colour. Thus, the dragon fruits are very bright in the images in C_r colour space and able to be segmented out. While using RGB colour space, the images show under Red plane show the bright object including sky, white object and dragon fruit. Thus, the images obtained from the Red plane will be threshold into binary images. All bright objects in the image will become white object with thresholding of 0.7. The red, green and blue colours have high correlation in RGB colour space. The dragon fruit is still able to be segmented out from the background with red channel minus green channel ($R-G$) shows in figure 4.1.

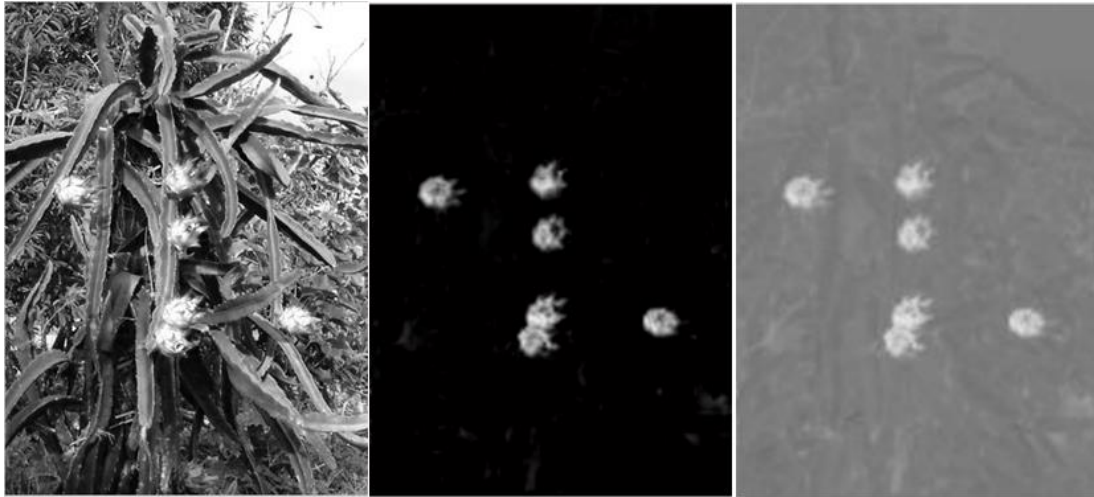






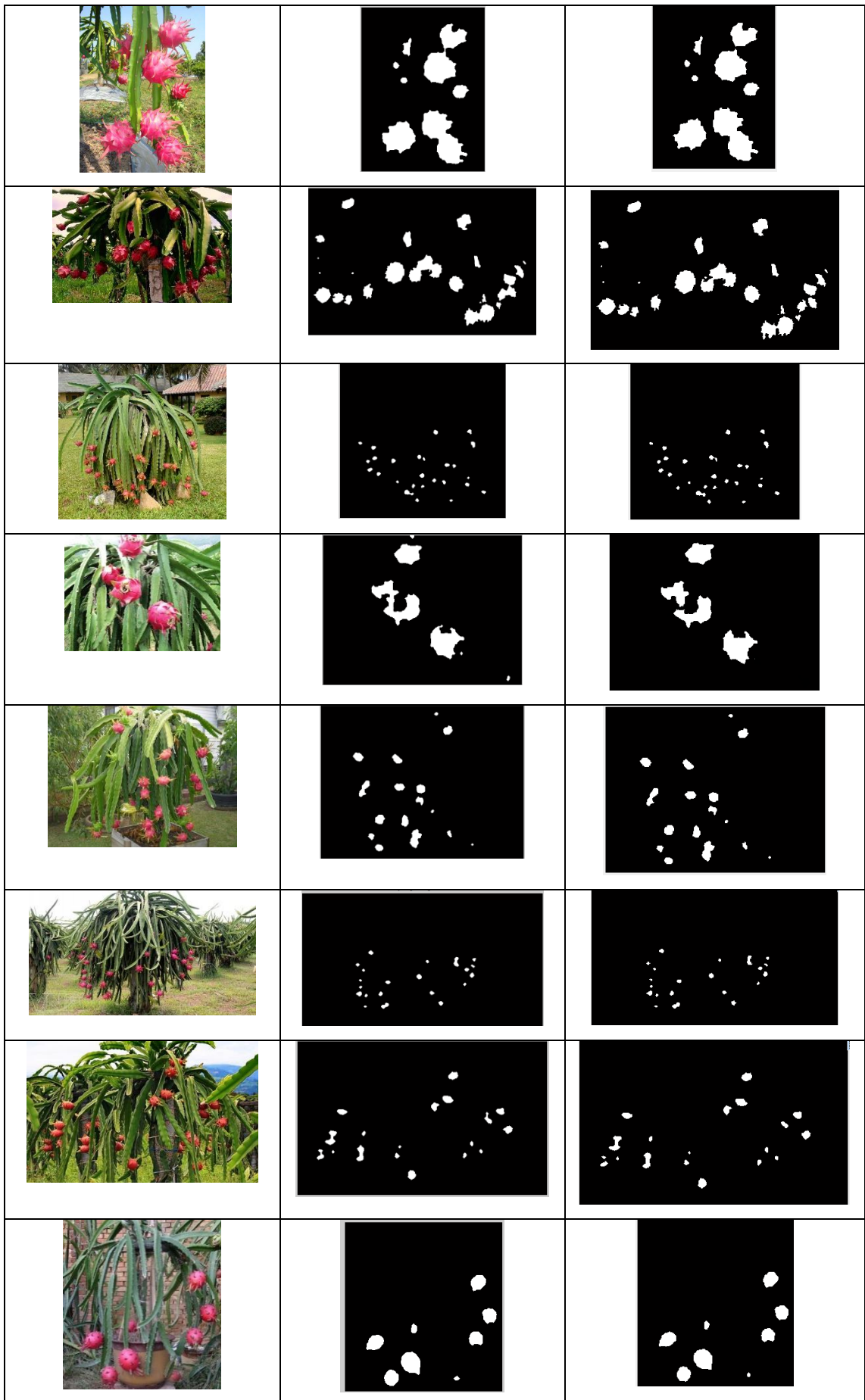


Figure 4.1: The colour Space thresholding comparison(Red channel, R-G channel and C_r channel)

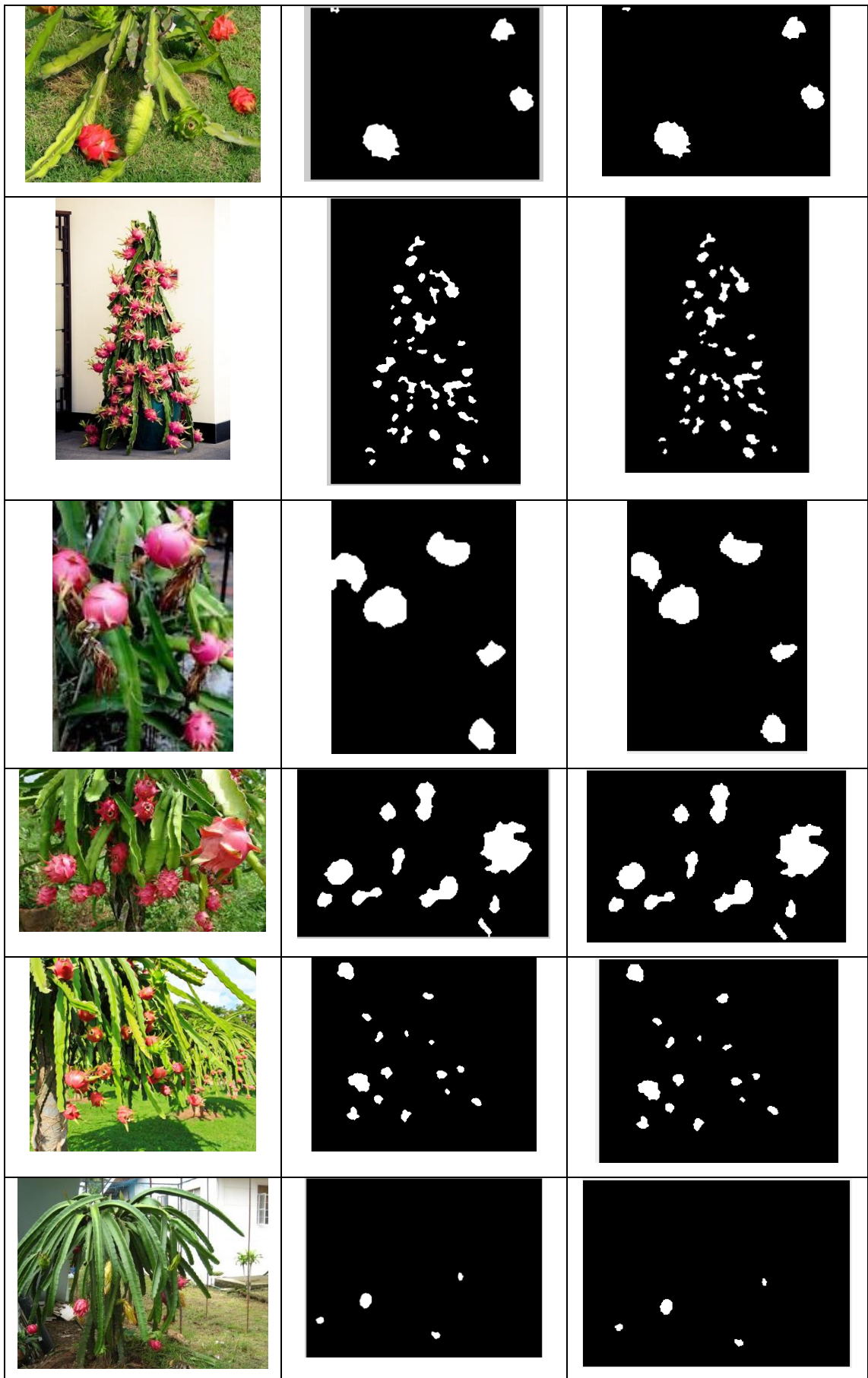
Based on the segmented images result from table 4.1, the YCbCr colour space is better than the RGB colour space since the YCbCr colour space C_r component only shows the reddish object brightly compared the Red plane in R-G colour space has more noise. The image segmentation based on YCbCr colour space will have a better segmentation result compared to the RGB colour space result. In RGB chrominance and luminance component are mixed that is why RGB is not chosen for colour analysis and colour based segmentation algorithm.

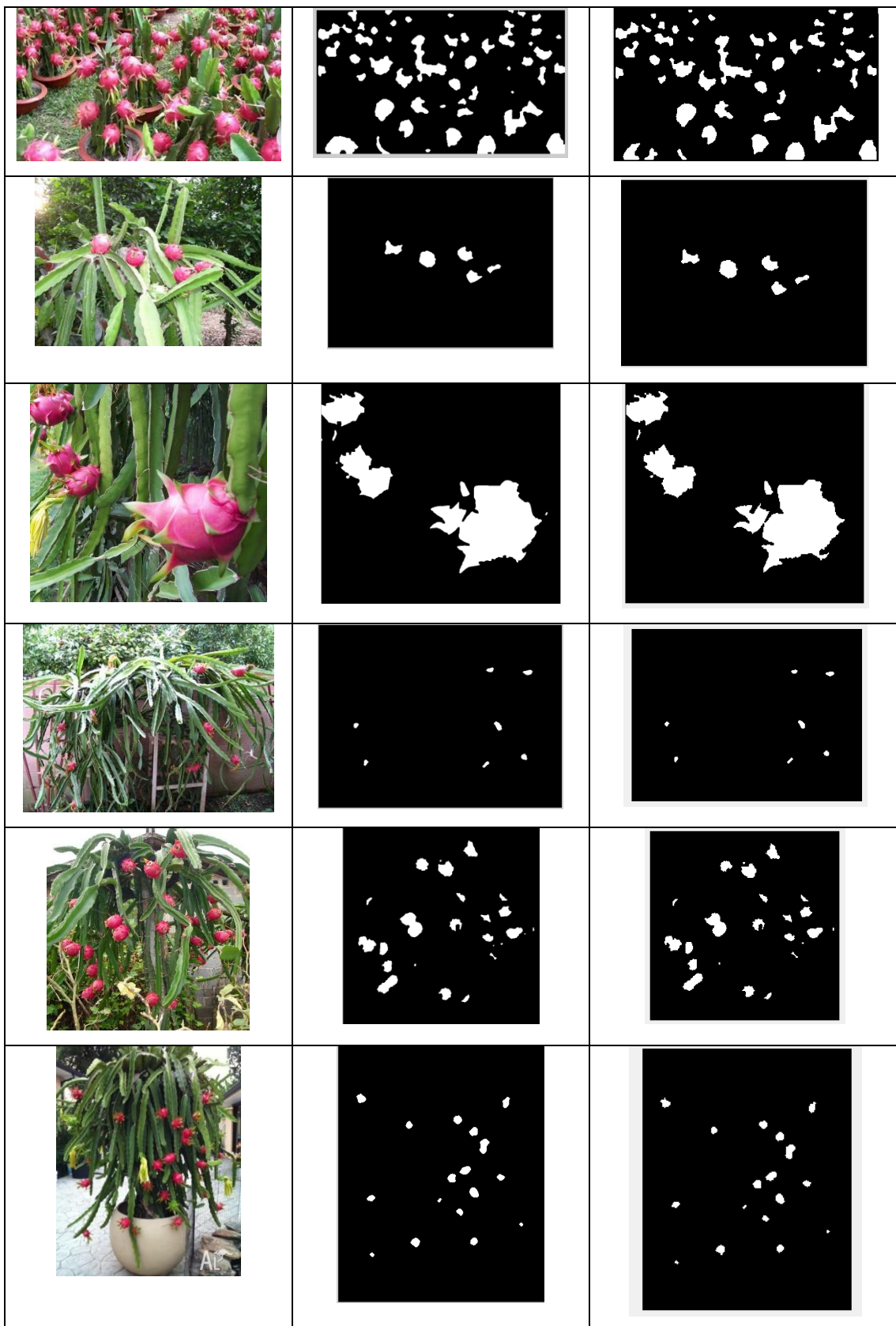
Table 4.1: The input images and segmented images in RGB and YCbCr colour space.

Input Image	Segmented Image(R-G)	Segmented Image(C_r)
		
		



















4.2 Result for Dragon Fruit

4.2.1 Automated Dragon Fruit Counting result for RGB and YCbCr colour space







Table 4.2 illustrates differences between automated counting in RGB colour space with red channel minus green channel(R-G) and automated counting in YCbCr colour space (Cr channel). A total number of 28 images are used in this project, the highest percentage error for automated counting in this algorithm for RGB and YCbCr colour space are 30% and 28.57% respectively due to many of the dragon fruits are occluded in the image. The average percentage error for RGB and YCbCr colour space automated counting are 13.5732% and 11.7614% respectively. The R^2 obtained for RGB is 0.792 from figure 4.1 and 0.997 for YCbCr from figure 4.2. The R^2 is a statistical measurement of the relationship of the data to the fitted regression line also known as the coefficient of determination. The YCbCr has a better automated counting result compared to the RGB result from table 4.2.







Table 4.2: Comparison of counting result for dragon fruit (colour space) to manual counting.

Input Image	Manual counting	Automated counting (R-G)	Percentage Error (R-G)	Automated counting (Cr)	Percentage Error (Cr)
	1	1	0.00	1	0.00
	3	3	0.00	3	0.00

	9	7	22.22	8	11.11
	24	18	25.00	20	16.67
	30	26	13.33	26	13.33
	4	5	25.00	4	0.00
	18	16	11.11	16	11.11
	25	21	16.00	21	16.00
	25	19	28.00	19	24.00
	8	8	0.00	8	0.00

	49	40	22.50	40	22.50
	2	2	0.00	2	0.00
	6	5	16.67	5	16.67
	7	7	0.00	7	0.00
	6	6	0.00	6	0.00
	20	16	20.00	15	25.00

	4	4	0.00	4	0.00
	51	37	27.45	37	27.45
	5	5	0.00	5	0.00
	14	10	28.57	10	28.57
	17	15	11.76	16	5.88
	4	4	20.00	4	20.00

	52	50	3.85	49	5.77
	5	5	0.00	5	0.00
	5	4	20	4	20
	8	7	12.50	7	12.50
	30	21	30.00	22	26.67
	23	17	26.09	17	26.09

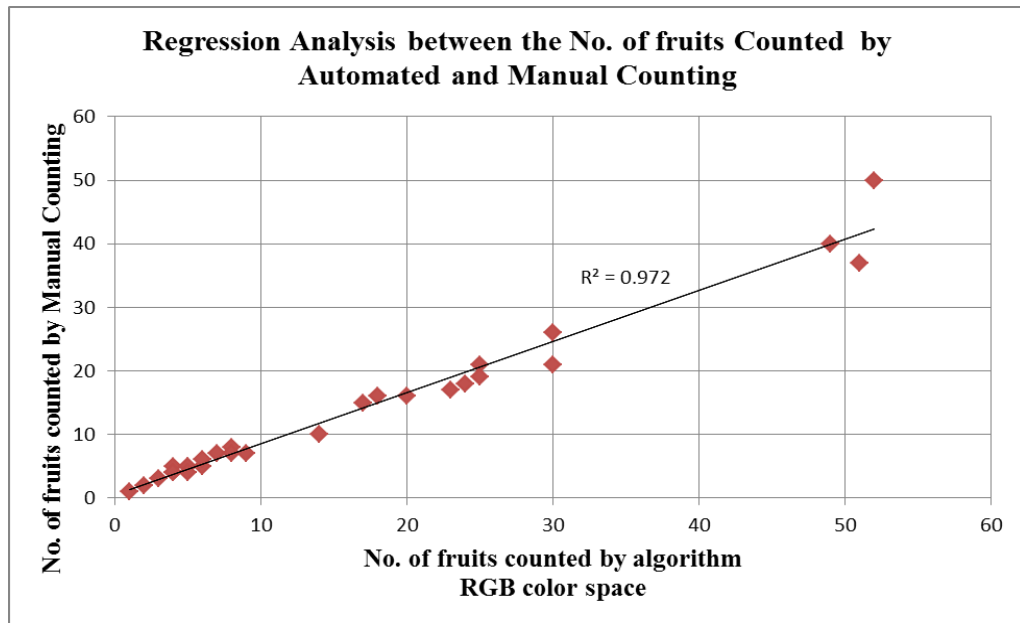


Figure 4.2: Regression analysis between the no. of fruits counted by RGB automated and manual counting.

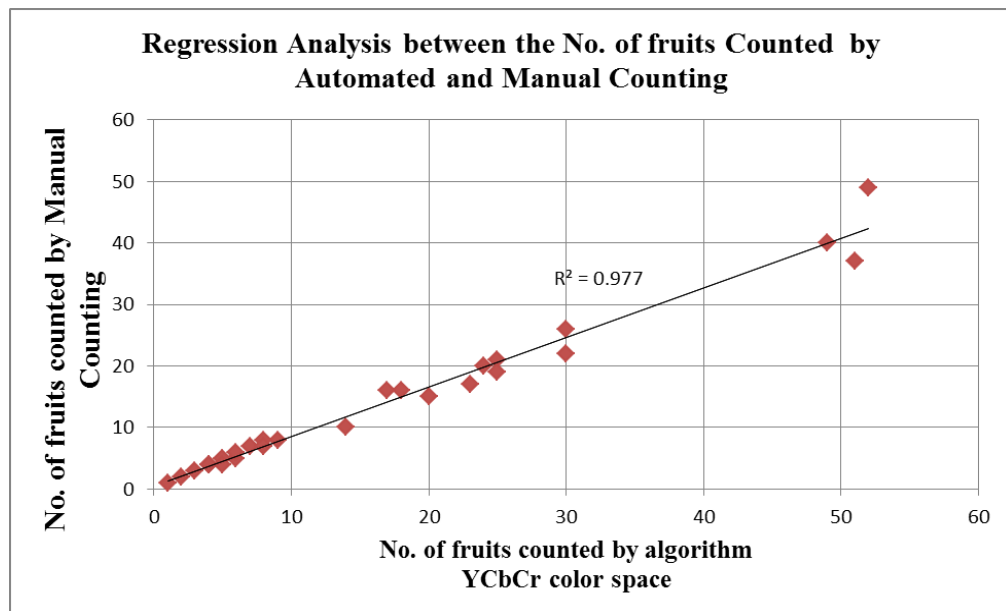














Figure 4.3: Regression analysis between the no. of fruits counted by YCbCr automated and manual counting.







4.2.2 Automated Dragon Fruit Counting result for RGB and YCbCr colour space and shape analysis.









Table 4.3 illustrates the input images and the differences between automated counting in RGB which is R-G channel with shape analysis and automated counting in YCbCr which is Cr channel with shape analysis. The highest percentage error for RGB and YCbCr automated counting in this algorithm are 29.41% and 21.74% respectively. The shape analysis determines the roundness value of the region of interest (ROI), any ROI with the roundness value between 0.45 and 0.60 will be classified as 2 dragon fruit found. Thus, the average percentage error for RGB is improved to 8.75%, while YCbCr improved to 8.0779%. The YCbCr colour space combined shape analysis has the best performance compared to previous algorithms.



Table 4.3: Comparison of counting result for dragon fruit (colour space + shape analysis) to manual counting.

Input Image	Manual counting	(R-G)+ shape	Percentage error (R-G) +shape	Cr+ shape	Percentage error (Cr+ shape)
	1	1	0.00	1	0.00
	3	3	0.00	3	0.00
	9	9	0.00	8	11.11

	24	20	16.67	27	12.5
	30	22	26.67	26	13.33
	6	7	16.67	5	16.67
	18	17	5.56	17	5.56
	25	22	12.00	21	16.00
	25	24	4.00	24	4.00
	8	8	0.00	8	0.00
	49	51	4.08	52	6.12
	2	2	0.00	2	0.00

	6	7	16.67	6	0.00
	7	8	14.29	8	14.29
	6	7	16.67	7	16.67
	20	22	10.00	21	5.00
	4	4	0	4	0
	51	36	29.41	40	17.65

	5	5	0.00	5	0.00
	14	14	0	13	7.14
	17	18	5.88	17	0
	4	4	0.00	4	0.00
	52	57	9.62	62	19.23
	5	6	20.00	6	20.00
	5	5	0.00	5	0.00
	8	7	12.80	7	12.50

	30	32	6.67	32	6.67
	23	19	17.39	18	21.74

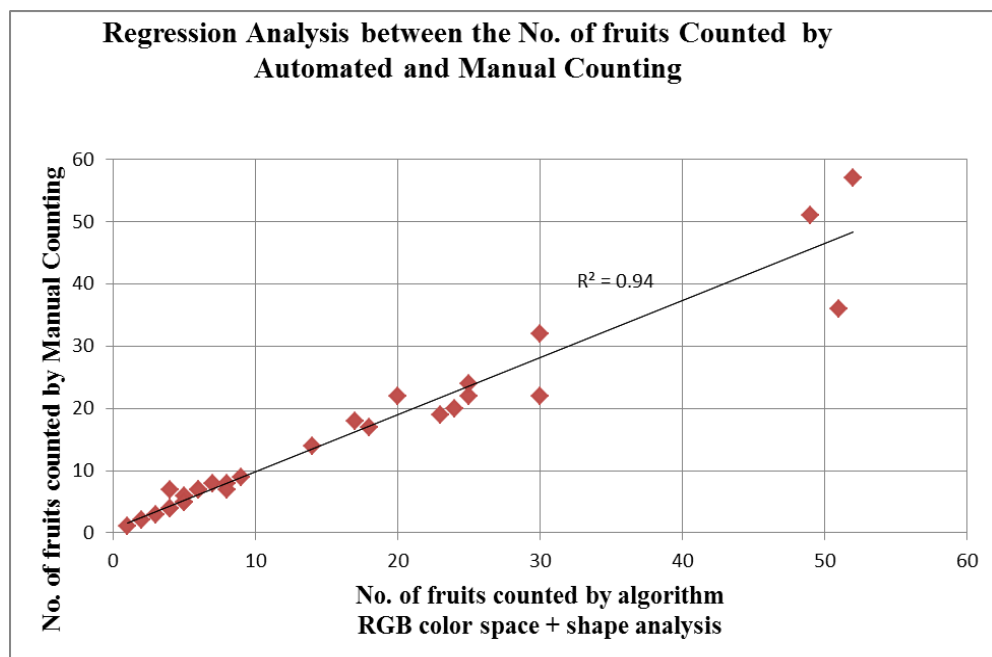


Figure 4.4: Regression analysis between the no. of fruits counted by RGB combined shape analysis automated and manual counting.

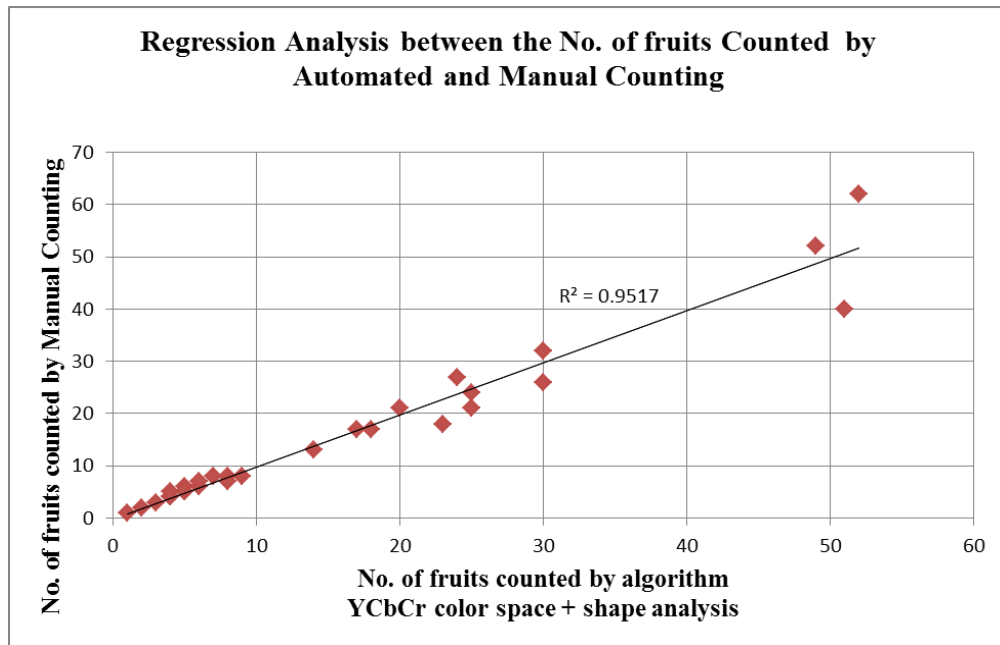


Figure 4.5: Regression analysis between the no. of fruits counted by YCbCr combined shape analysis automated and manual counting.

4.2.3 Summary of Automated Dragon Fruit Counting

The RGB(R-G) have the highest percentage error followed by YCbCr(Cr), RGB(R-G) combined shape analysis. The shape analysis slightly improved the accuracy of the algorithm developed in both colour spaces. From table 4.4, the YCbCr colour space (Cr) combined shape analysis have the best result of counting the dragon fruit in digital still images with percentage error of 8.0779% which the R^2 value of 0.9517.

Table 4.4: The summary of the results obtained from different colour space and shape analysis.


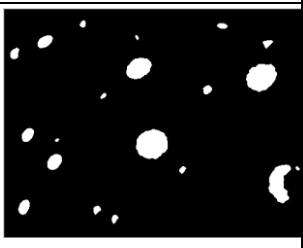
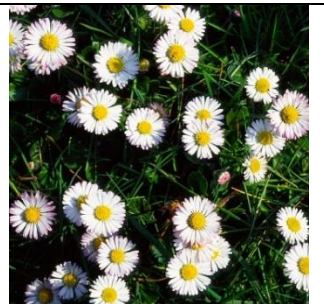
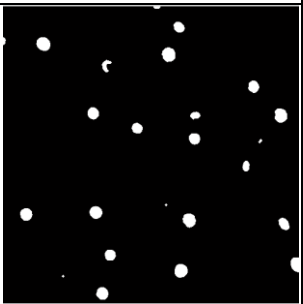

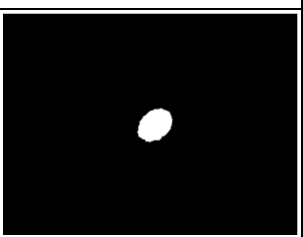
	RGB (R-G)	YCbCr (Cr)	RGB (R-G) + Shape analysis	YCbCr (Cr) + Shape analysis
Average percentage error	13.5732%	11.7614%	8.7518%	8.0779%
R^2 in Regression Analysis	0.972	0.977	0.9400	0.9517


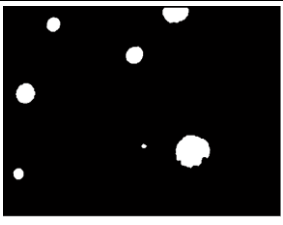

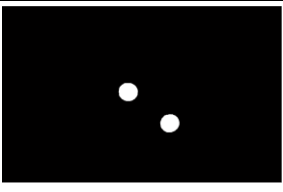

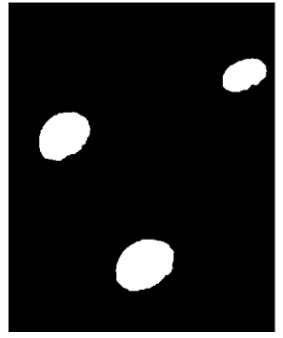





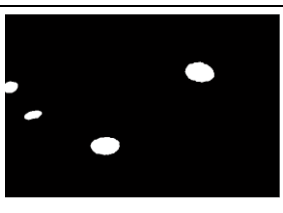

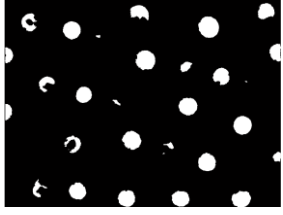
4.3 Result for Daisy




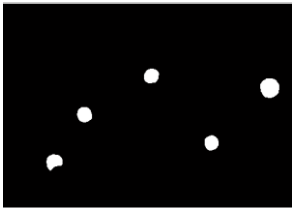

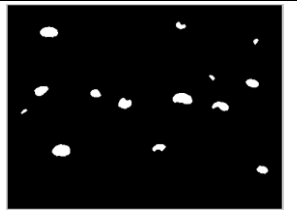





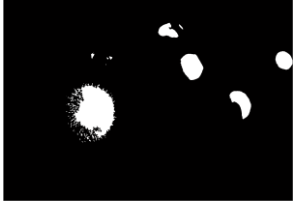

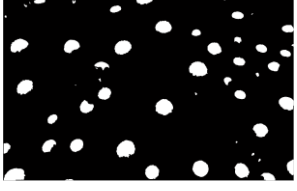


4.3.1 Daisy Segmentation and Automated Counting result for RGB(R-B) colour space


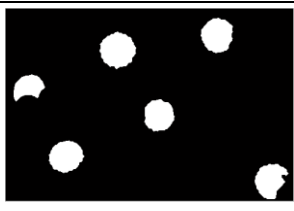

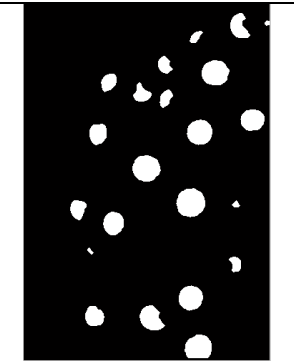

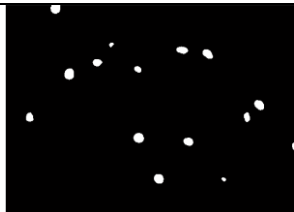



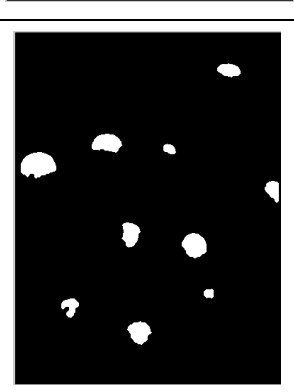
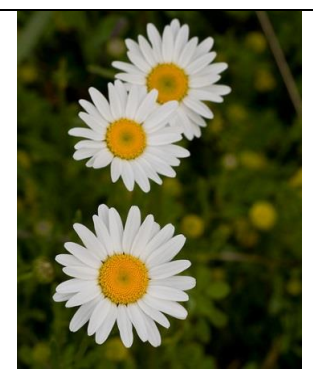
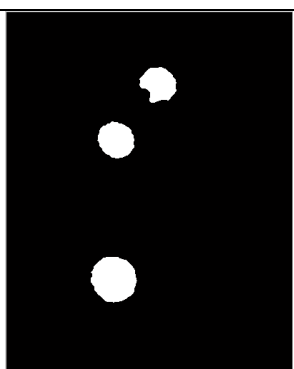
Table 4.5 illustrates differences between automated counting in RGB colour space which red channel minus blue channel(R-B). A total number of 26 daisy images used in this project, the highest percentage error for automated counting in this algorithm for RGB colour space is 27.27% due to illumination of the input image. The average percentage error for RGB colour space automated counting is 11.30% and value of R^2 is 0.9348 from figure 4.5.





Table 4.5: Comparison of counting result for daisy flower (colour space) to manual counting.

Input Image	Segmented Binary Image	Manual counting	Automated counting	Percentage Error
		19	20	5.36
		28	24	14.29
		1	1	0.00

		7	7	0.00
		2	2	0.00
		3	3	0.00
		37	33	10.81
		20	23	15.00
		4	4	0.00
		32	27	15.63

		3	3	0.00
		6	5	16.67
		19	14	26.32
		12	7	41.67
		7	7	0.00
		6	5	16.67
		41	50	21.95
		1	1	0.00

		6	6	0.00
		20	22	10.00
		18	15	16.67
		28	24	14.29
		12	10	16.67
		3	3	0.00

		11	14	27.27
		9	8	11.11

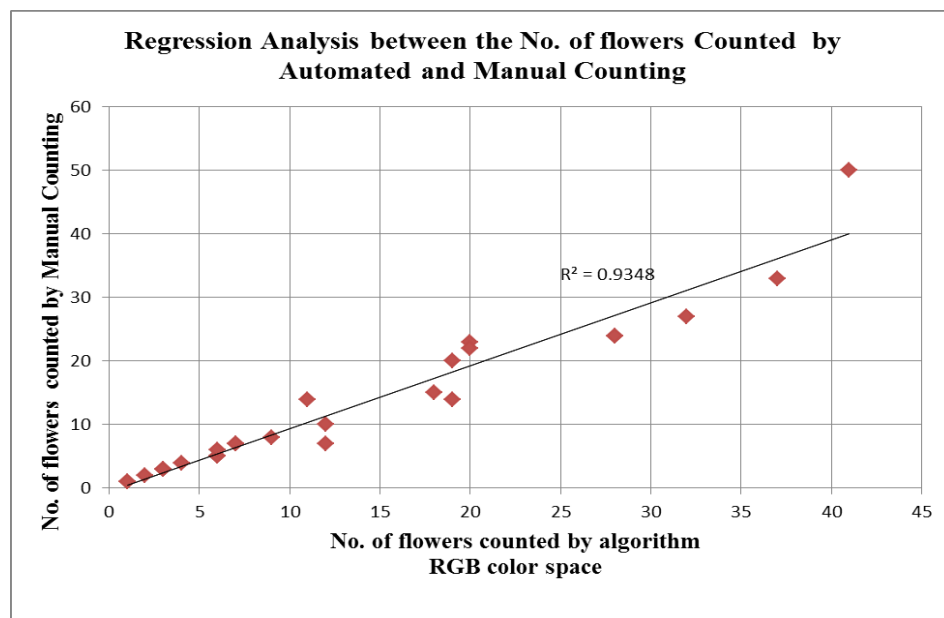



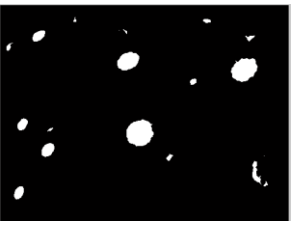







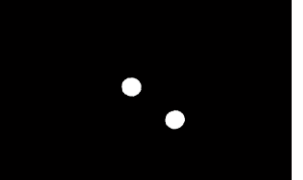
Figure 4.6: Regression analysis between the no. of flowers counted by RGB automated and manual counting.


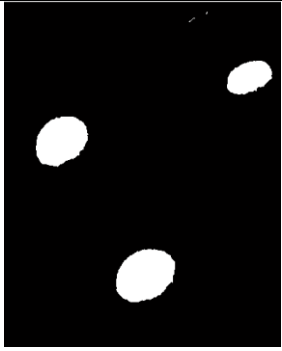

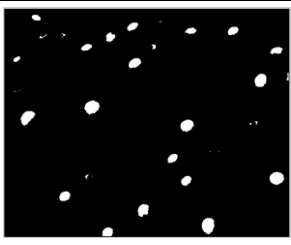

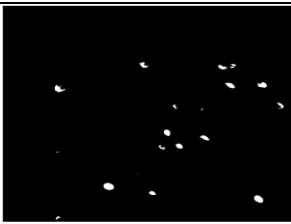

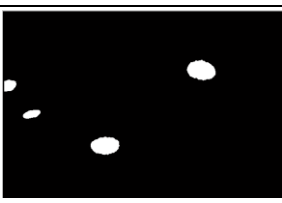

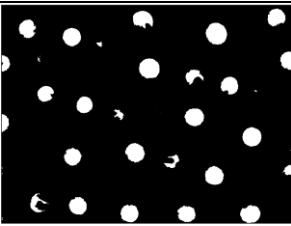



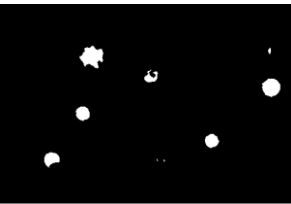
4.3.2 Daisy Segmentation and Automated Counting result for YCbCr(Cr) colour space


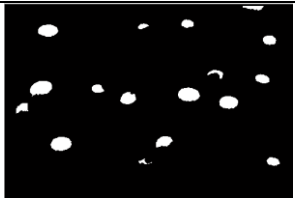

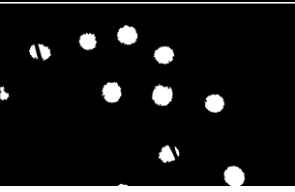





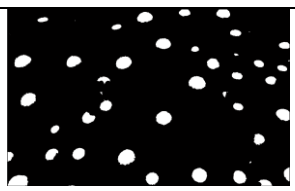

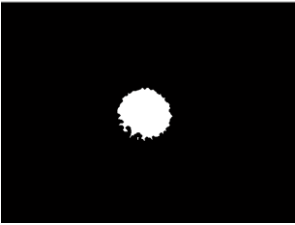

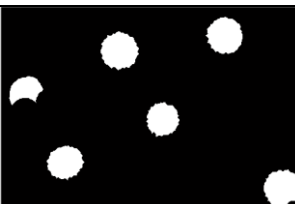
Table 4.6 illustrates differences between automated counting in YCbCr(Cr) colour space and manual counting. A total number of 26 daisy images used in this project, the highest percentage error for automated counting in this algorithm for YCbCr colour space is 24.32% due to illumination of the input image and overlapping


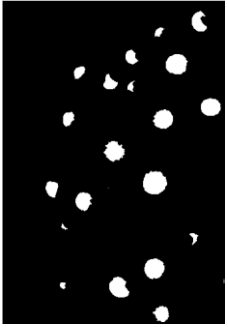

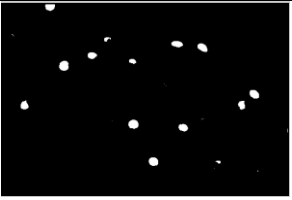

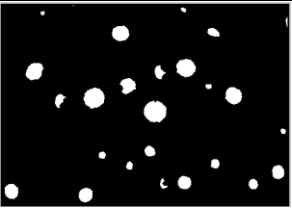

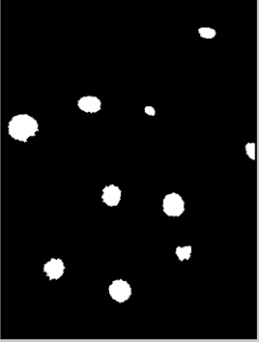

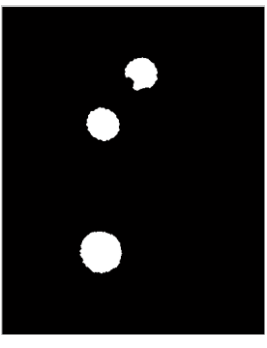

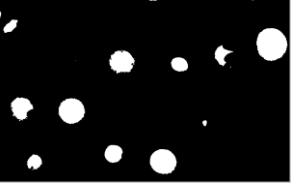
flower. The average percentage error for YCbCr colour space automated counting is 5.5434% and value of R^2 is 0.9751 from figure 4.6.


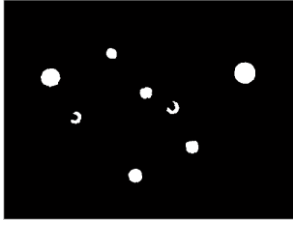
Table 4.6: The result obtained by the Automated counting (Cr) and compared to the manual counting.

Input Image	Segmented Binary Image	Manual counting	Automated counting	Percentage Error
		19	17	10.53
		28	26	7.14
		1	1	0.00
		7	7	0.00
		2	2	0.00

		3	3	0.00
		37	28	24.32
		20	18	10.00
		4	4	0.00
		32	28	12.50
		3	3	0.00
		6	7	16.67

		19	17	10.53
		12	13	8.33
		7	8	14.29
		6	6	0.00
		41	42	2.44
		1	1	0.00
		6	6	0.00

		20	20	0.00
		18	17	5.56
		28	25	10.71
		12	10	0.00
		3	3	0.00
		11	11	0.00

		9	8	11.11
---	---	---	---	-------

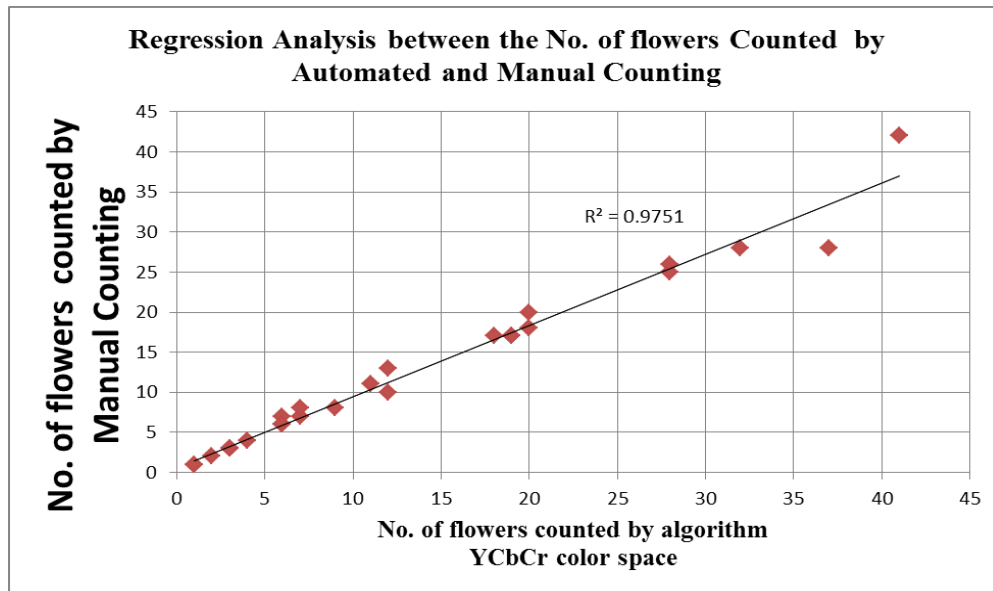


Figure 4.7: Regression analysis between the no. of flowers counted by YCbCr automated and manual counting

4.3.3 Summary of Daisy Automated Counting

Total number of 26 daisy images was used to perform the automated counting algorithm. From table 4.7, YCbCr colour space has a better performance compared to the RGB colour space has an average error of 5.5434%. The R^2 value of YCbCr colour space is higher than the RGB colour space which is 0.9751.

Table 4.7: The automated counting result and regression analysis result obtained from RGB and YCbCr colour space

	RGB colour space (R-B)	YCbCr colour space (Cr)
Average error (%)	11.30	5.5434
R^2	0.9348	0.9751

4.4 Limitation of fruit and flower automated counting

The algorithm developed for this project is not able to detect the fruits and flowers which are a dark place and under the shade. The algorithm only detects the fruit and flowers with controlled illumination and direction of sun light to obtain an accurate result by automated counting. The algorithm will falsely detect the interested colour and round objects as fruits or flowers, since the images are taken in the farm.

4.5 Yield Prediction

4.5.1 Dragon Fruit Yield Prediction

Dragon fruit is red in colour, round shape and size between 10-15 cm and weights between 300 to 500 grams. Dragon fruit can be harvested all year round and the peak seasons are around April and September, the harvesting time might vary in some places. The average production in one hectare is more than 10 tons per hectare; once the crops mature enough. There are about 1500 dragon fruit plants in one hectare. The size of fruit depends on several factors such as weather, sufficient water and farm management (Zainudin, 2015).









The yield of the dragon fruit is estimated to be 10tons per hectare and around 1500 dragon fruit plants in one hectare, one dragon fruit weight 400g as reference. From table 4.8, the predicted yield varies from 10140kg/hectare to 11160kg/hectare













based on the 5, 10, 15 and 20 dragon fruit tree samples. The 20 dragon fruit trees have the higher variation of dragon fruit found on each tree, the predicted yield is 10350kg/hectare compared to the references yield 10000kg/hectare has the percentage error of 3.5%. However, from comparison in table 4.8, 10images give least error that is around 10140kg/hectare with percentage error of 1.4%.

Table 4.8: The predicted dragon fruits yield

Dragon fruit trees used to predict the yield	5	10	15	20
Average fruit on one tree	18.6	16.9	17.7	17.25
Weight of dragon fruits on one tree (kg)	7.44	6.76	7.09	6.90
Yield (kg/hectare)	11160	10140	10635	10350

Table 4.9: The input images and automated counting result.

Input Image	Automated counting	Input Image	Automated counting
	32		18
	8		27
	24		8
	21		6

	8		21
	17		24
	4		7
	52		26
	13		5
	7		17

4.5.2 Daisy Yield Prediction






The spacing of daisy plant between the rows should be 30-40 cm and 25-30 cm within the row accommodating 8-10 plants/m² and harvests 3 times per year. Average yield of cut flowers under open condition are around 130-160 flowers/m²/year of which only 15-20 % of I grade quality in open conditions (Sandesh, 2010). The references yield takes the average of the flowers under open condition 145flowers/m²/year.











The predicted yield is based on total flowers in the images and the area in m^2 . The area estimated based on the flowers size on the image and is compare to estimate the actual size of the daisy images. The average predicted yield is 137.12 flowers/ m^2 /year compared to the reference yield which is 145 flowers/ m^2 /year. The predicted yield has the percentage error of 5.52% as shown in the table 4.10.

Table 4.10: The percentage error of the predicted yield for daisy

references yield (flowers/ m^2 /year)	predicted yield (flowers/ m^2 /year)	percentage error
145	137.12	5.52%

Table 4.11: The predicted yield of Daisy Flower

Input Images	Image size	Flower size	Area (m^2)	Total flower	Yield(flowers / m^2 /year)
	500x375	72x58	0.4490	28	126.96
	500x375	113x96	0.1728	7	121.53
	1920x1200	357x281	0.2297	12	156.72
	375x500	61x58	0.5300	12	67.92
	500x333	129x108	0.1195	7	175.74

	2560x1600	325x337	0.3740	11	88.23
	500x352	72x64	0.3819	28	219.96
	615x410	81x71	0.4384	18	123.18
	344x500	90x84	0.2275	20	175.82
	500x375	75x76	0.3289	9	82.08
	500x334	68x55	0.4465	19	127.65
	2560x1920	283x219	0.7931	20	75.66
	500x500	80x80	0.3906	28	215.04
	1280x960	153x108	0.7436	41	165.42
	1280x960	137x109	0.8229	37	134.88

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Nowadays the automated counting systems are in high demand in agricultural field. This project uses image processing techniques to segment out the region of interest and count the dragon fruit and daisy (fruit and flower) in digital still images, then predict the yield of the dragon fruit and daisy. In this project, the implemented image processing techniques include colour, size and shape features. The more image processing techniques apply in the algorithms developed in this project, the higher the accuracy of the segmentation of region of interest.

The proposed algorithm is able to segment fruit and flower (dragon fruit and daisy) and quantify total number of fruit and flower with an average error of 8.0779% and 5.5434% with the R^2 value of 0.9517 and 0.9751 respectively. The predicted yield for dragon fruit is 10140kg/hectare, while the daisy flower is 137flowers/m²/year with the percentage error of 1.40% and 5.52% respectively. This is a simple and less costly method may help farmer to predict the yield of the farm and able to arrange the transportation and sale of the harvested products to make more profit.

5.2 Recommendations

An advanced automated fruit and flower counting system shall be developed in future in to order to further boost the accuracy of segmentation and the predicted yield result. There are some extra features and improvement can be added into the algorithm developed such as unsupervised learning and surface texture feature to increase the region of interest segmentation accuracy. A robot mounted with camera can be designed to move around the farm and automated predict the yield of farm.

REFERENCES

- Anisha, S., Divya, G., Shanu, S. 2013. A Survey of Computer Vision Methods for Counting Fruits and Yield Prediction. *International Journal of Computer Science Engineering (IJCSE)* Vol. 2 No.06, pp.346-350.
- Bala. A. 2012. An Improved Watershed Image Segmentation Technique using MATLAB. *International Journal of Scientific & Engineering Research* 2012. [online] Available at: <<http://www.ijser.org/researchpaper%5CAn-Improved-Watershed-Image-Segmentation-Technique-using-MATLAB.pdf>> [Accessed 15 March 2015].
- C. Hung, J. Underwood, J. Nieto, and S. Sukkarieh., 2013. A Feature Learning Based Approach for Automated Fruit Yield Estimation. In: *IEEE, Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference*. Tokyo, 3-7 November 2013.
- D, Choi., W. S. Lee., Reza, E., 2013. Detecting and counting citrus fruit on the ground using machine vision. In: *ASABE, 2013 ASABE Annual International Meeting*. Kansas City, Missouri, 21st -24th July 2013. Missouri: ASABE. Available at: <abe.ufl.edu/precag/pdf/2013DaeunASABE.pdf> [Accessed 28 January 2015]
- Dorj, U.O., Lee, K.K., and Lee, M. 2013. A computer Vision Algorithm for Tangerine Yield Estimation. *International Journal of Bio-Science and Bio-Technology*, Vol.5, No.5 pp. 101-110.
- Hamirul A.H., Sharifah L. S. A., Nursuriati J., Hazaruddin H. 2014. A Rule-based Segmentation Method for Fruit Images under Natural Illumination. In: *International Conference on Computer, Control, Informatics and Its Applications*, 2014. Bandung, Indonesia. 21-23 Oct. 2014. Available at: <<http://ieeexplore.ieee.org.libezp.utar.edu.my/stamp/stamp.jsp?tp=&arnumber=7042593>> [Accessed 23 March 2015]
- K. Sandesh., 2010. Agriculture in India. GERBERA [online] Available at: <<http://www.krishisandesh.com/flowers/gerbera.>> [Accessed 31 August 2015].

- M.J.C.S. Reis, R. Morais, E. Peres, C. Pereira, O. Contente, S. Soares, A. Valente, J. Baptista, P.J.S.G. Ferreira f, J. Bulas Cruz 2012. Automatic detection of bunches of grapes in natural environment from color images. *Journal of Applied Logic*. 10(4), pp. 285-290.
- Patel H. N., Jain R.K., Joshi M. V., 2012. Automatic Segmentation and Yield Measurement of Fruit using Shape Analysis. 2012 *International Journal of Computer Applications*, [online] 45(7), pp19-24. Available at: <<http://research.ijcaonline.org/volume45/number7/pxc3879119.pdf>> [Accessed 28 March 2015].
- Q. Wang., Stephen, N., Marcel, B., Sanjiv. S., 2012. Automated Crop Yield Estimation for Apple Orchards. In: 13th International Symposium on Experimental Robotics (ISER 2012). Quebec City, July 2012. Quebec City: ISER Available at: <http://www.ri.cmu.edu/casc/publications/yieldEstimationISER_2012.pdf> [Accessed 28 January 2015]
- Sarkate, R.S., Kalyankar, N.V., Khanale, P.B., 2013. Application of computer vision and color image segmentation for yield prediction precision. In: 2013 International Conference on Information Systems and Computer Networks (ISCON). Mathura, India 9-10 March 2013. Mathura: IEEE. Available at: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6524164>> [Accessed 14 March 2015]
- Wijethunga, P., Samarasinghe, S., Kulasiri, D., Woodhead, I. 2008. Digital Image Analysis Based Automated Kiwifruit Counting Technique. In: *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*. Christchurch, 26-28 Nov. 2008 New Zealand: IEEE.
- Woo, C.S., Mirisae, S.H., 2009. A New Method for Fruits Recognition System. In: *Electrical Engineering and Informatics 2009 international conference*. Selangor, 5th August 2009. Selangor: IEEE. Available at: <http://www.mncc.com.my/tranx/Woo%20CS%20FruitRecognition_journal.pdf> [Accessed 26 January 2015]
- Y. Song., C.A. Glasbey., G.W. Horgan., G. Polder., J.A. Dieleman., G.W.A.M. van der Heijden., 2013. Automatic fruit recognition and counting 1 from multiple images. Available at: <<https://www.bioss.ac.uk/people/chris/fruit.pdf>> [Accessed 28 January 2015]
- Yin. J., Mao. H., Xie. Y. 2008. Segmentation Methods of Fruit Image and Comparative Experiments. In: *2008 International Conference on Computer Science and Software Engineering*. Wuhan, Hubei. 12-14 Dec. 2008. Wuhan: IEEE Available at: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4721944&isnumber=4721668>> [Accessed 28 March 2015].

Zainudin. M., 2015. Dragon Fruits Cultivation in Malaysia. [online] Available at: <<http://www.scribd.com/doc/91726161/Dragon-Fruits-Cultivation-in-Malaysia#scribd>> [Accessed 18 August 2015].

APPENDICES

APPENDIX A: Computer Programme Listing

(a) Matlab coding for automated dragon fruit counting (R-G)

```
[File_Name, Path_Name] = imgetfile(); %read the image user selected
fruit= imread(File_Name); % assign the image to 'fruit'
imshow(fruit), title('Input Image'); %Show the input image

Red=fruit(:,:,1); % red channel
Green=fruit(:,:,2);% green channel
object=imsubtract(Red,Green);%red- green channel
figure, imshow(object);
f=fspecial('gaussian', 3,3); %Define 3x3 gaussian filter
Filtered=imfilter(object, f); %Filter the image
%Convert into binary image with threshold of 0.2
bw1=im2bw(Filtered, 0.2);
imshow(Filtered), title('Redchrominance Image'); % Show Filtered image
Holescleared = imfill(bw1,'holes'); %Remove holes in the binary image
mask=strel('disk',3); % Disk shape mask with size of 3
Image_erosion=imerode(Holescleared,mask); %Erode image with diamond mask
mask2=strel('disk',3); % Disk shape mask with size of 3
mask1=strel('disk',2); % Disk shape mask with size of 2
Opening = imopen(Image_erosion,mask1);% Opening of image with square mask
Closing = imclose(Opening,mask2); %Closing of image with diamond mask
remove= bwareaopen(Closing,20); %Remove small object in image
figure,
imshow(remove), title('Binary Image of Segmented Fruit');

[L,n] = bwlabel(remove);
% to count the area of the region
fruit_region = regionprops(L,'Area','BoundingBox');
fruit_area = [fruit_region.Area];
% find the regions that greater than 1% of total segmented region
fruit_idx = find(fruit_area>(0.01)*max(fruit_area));
fruit_shown = ismember(L,fruit_idx);

for n = 1:length(fruit_idx) % Create a outline of the region
```

```

idx =fruit_idx(n);
fruit_region(idx).BoundingBox;
xmin = round(fruit_region(idx).BoundingBox(1));
ymin = round(fruit_region(idx).BoundingBox(2));
xmax      =      round(fruit_region(idx).BoundingBox(1)      +
fruit_region(idx).BoundingBox(3));
ymax      =      round(fruit_region(idx).BoundingBox(2)      +
fruit_region(idx).BoundingBox(4));
end

S = regionprops(remove , 'Centroid'); %find the centroid of fruit
centroids = cat( 1, S.Centroid );
figure,
imshow (fruit), title('Detection Result'),
    %Show the total Dragon fruit found
    xlabel(['Number of Dragon fruit found:',num2str(length(fruit_idx))]);
hold on
%Plot the centroid with blue marker size of 10
plot (centroids(:,1), centroids(:,2),'b+', 'MarkerSize',10);
hold off

```

(b) Matlab coding for automated dragon fruit counting (Cr)

```

[File_Name, Path_Name] = imgetfile(); %read the image user selected
fruit= imread(File_Name); % assign the image to 'fruit'
imshow(fruit), title('Input Image'); %Show the input image
k= rgb2ycbcr(fruit); %Convert RGB to YCbCr
Redchrominance=k(:, :,3); %Shown Cr
f=fspecial('gaussian', 3,3); %Define 3x3 gaussian filter
Filtered=imfilter(Redchrominance, f); %Filter the Cr image
%Convert into binary image with threshold of 0.6
bw1=im2bw(Filtered, 0.6);
imshow(Filtered), title('Redchrominance Image'); % Show Filtered image
Holescleared = imfill(bw1,'holes'); %Remove holes in the binary image
mask=strel('disk',3); % Disk shape mask with size of 3
Image_erosion=imerode(Holescleared,mask); %Erode image with diamond mask
mask2=strel('disk',3); % Disk shape mask with size of 3
mask1=strel('disk',2); % Disk shape mask with size of 2
Opening = imopen(Image_erosion,mask1);% Opening of image with square mask
Closing = imclose(Opening,mask2); %Closing of image with diamond mask
remove= bwareaopen(Closing,20); %Remove small object in image
figure,
imshow(remove), title('Binary Image of Segmented Fruit');

[L,n] = bwlabel(remove);
% to count the area of the region
fruit_region = regionprops(L,'Area','BoundingBox');
fruit_area = [fruit_region.Area];

```

```

% find the regions that greater than 1% of total segmented region
fruit_idx = find(fruit_area>(0.01)*max(fruit_area));
fruit_shown = ismember(L,fruit_idx);

for n = 1:length(fruit_idx) % Create a outline of the region
    idx =fruit_idx(n);
    fruit_region(idx).BoundingBox;
    xmin = round(fruit_region(idx).BoundingBox(1));
    ymin = round(fruit_region(idx).BoundingBox(2));
    xmax      =      round(fruit_region(idx).BoundingBox(1)      +
fruit_region(idx).BoundingBox(3));
    ymax      =      round(fruit_region(idx).BoundingBox(2)      +
fruit_region(idx).BoundingBox(4));
end

S = regionprops(remove , 'Centroid'); %find the centroid of fruit
centroids = cat( 1, S.Centroid );
figure,
imshow (fruit), title('Detection Result'),
    %Show the total Dragon fruit found
    xlabel(['Number of Dragon fruit found:',num2str(length(fruit_idx))]);
hold on
    %Plot the centroid with blue marker size of 10
plot (centroids(:,1), centroids(:,2),'b+', 'MarkerSize',10);
hold off

```

(c) Matlab coding for automated dragon fruit counting (R-G combined shape analysis)

```

[File_Name, Path_Name] = imgetfile(); %read the image user selected
fruit= imread(File_Name); % assign the image to 'fruit'
figure,
imshow(fruit), title('Input Image'); %Show the input image
Red=fruit(:,:,1); %red channel
Green=fruit(:,:,2);%green channel
object=imsubtract(Red,Green);%Red minus green channel
figure, imshow(object);
f=fspecial('gaussian', 3,3); %Define 3x3 gaussian filter
Filtered=imfilter(object, f); %Filter the image
%Convert into binary image with threshold of 0.2
bw1=im2bw(Filtered, 0.2);
count = 0;
Holescleared = imfill(bw1,'holes'); %Remove holes in the binary image
mask=strel('disk',2); % Disk shape mask with size of 2
Image_erosion=imerode(Holescleared,mask); %Erode image with diamond mask
figure,
imshow(Image_erosion);
mask2=strel('disk',3); % Disk shape mask with size of 3
mask1=strel('disk',2); % Disk shape mask with size of 2
Opening = imopen(Image_erosion,mask1);% Opening of image with square mask

```



```

Closing = imclose(Opening,mask2); %Closing of image with diamond mask
remove= bwareaopen(Closing,20); %Remove small object in image
figure,
imshow(remove), title('Binary Image of Segmented Fruit');

[L,n] = bwlabel(remove);
% to count the area of the region
fruit_region = regionprops(L,'Area','BoundingBox');
fruit_area = [fruit_region.Area];
% find the regions that greater than 1% of total segmented region
fruit_idx = find(fruit_area>(0.01)*max(fruit_area));
fruit_shown = ismember(L,fruit_idx);
[B,L] = bwboundaries(remove, 'noholes');
% Display the label matrix and draw each boundary
imshow(fruit);
hold on

stats = regionprops(L,'Area','Centroid');
threshold = 5;
threshold1 = 0.1;
% loop over the boundaries
for k = 1:length(B)
    % obtain (X,Y) boundary coordinates corresponding to label 'k'
    boundary = B{k};
    % compute a simple estimate of the object's perimeter
    delta_sq = diff(boundary).^2;
    perimeter = sum(sqrt(sum(delta_sq,2)));
    % obtain the area calculation corresponding to label 'k'
    area = stats(k).Area;
    % compute the roundness metric
    metric = 4*pi*area/perimeter^2;
    % display the results
    metric_string = sprintf('%2.2f',metric);
    % mark objects above the threshold with a black circle
    if (metric < threshold && metric > threshold1)
        boundary = B{k};
        plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2);
        text(boundary(1,2)-35,boundary(1,1)+13,metric_string,'Color','r',...
            'FontSize',14,'FontWeight','bold');
    End

    % Count the segmented region with metric between 0.45 and 0.60 as 2fruit
    if (metric < 0.60 && metric > 0.45)
        boundary = B{k};
        plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2);
        text(boundary(1,2)-35,boundary(1,1)+13,metric_string,'Color','r',...
            'FontSize',14,'FontWeight','bold');
        count= count+1;
    end
end
end

```

```

count=(count+(length(fruit_idx)));
    %Show the total Dragon fruit found
    xlabel(['Number of Dragon fruit found:',num2str(count)]);

```

(d) Matlab coding for automated dragon fruit counting (Cr combined shape analysis)

```

[File_Name, Path_Name] = imgetfile(); %read the image user selected
fruit= imread(File_Name); % assign the image to 'fruit'
figure,
imshow(fruit), title('Input Image'); %Show the input image
k= rgb2ycbcr(fruit); %Convert RGB to ycbcr
Redchrominance=k(:, :,3); %Shown redchrominance
f=fspecial('gaussian', 3,3); %Define 3x3 gaussian filter
Filtered=imfilter(Redchrominance, f); %Filter the Cr image
count = 0;
%Convert into binary image with threshold of 0.6
bw1=im2bw(Filtered, 0.6);
imshow(Filtered), title('Redchrominance Image'); % Show Filtered image
Holescleared = imfill(bw1,'holes'); %Remove holes in the binary image
mask=strel('disk',2); % Disk shape mask with size of 2
Image_erosion=imerode(Holescleared,mask); %Erode image with diamond mask
figure,
imshow(Image_erosion);
mask2=strel('disk',3); % Disk shape mask with size of 3
mask1=strel('disk',2); % Disk shape mask with size of 2
Opening = imopen(Image_erosion,mask1);% Opening of image with square mask
Closing = imclose(Opening,mask2); %Closing of image with diamond mask
remove= bwareaopen(Closing,20); %Remove small object in image
figure,
imshow(remove), title('Binary Image of Segmented Fruit');
[L,n] = bwlabel(remove);
% to count the area of the region
fruit_region = regionprops(L,'Area','BoundingBox');
fruit_area = [fruit_region.Area];
% find the regions that greater than 1% of total segmented region
fruit_idx = find(fruit_area>(0.01)*max(fruit_area));
fruit_shown = ismember(L,fruit_idx);

[B,L] = bwboundaries(remove, 'noholes');

% Display the label matrix and draw each boundary
imshow(fruit);
hold on
stats = regionprops(L,'Area','Centroid');

threshold = 5;
threshold1 = 0.1;
% loop over the boundaries

```

```

for k = 1:length(B)
    % obtain (X,Y) boundary coordinates corresponding to label 'k'
    boundary = B{k};
    % compute a simple estimate of the object's perimeter
    delta_sq = diff(boundary).^2;
    perimeter = sum(sqrt(sum(delta_sq,2)));
    % obtain the area calculation corresponding to label 'k'
    area = stats(k).Area;
    % compute the roundness metric
    metric = 4*pi*area/perimeter^2;
    % display the results
    metric_string = sprintf('%2.2f',metric);
    % mark objects above the threshold with white circle
    if (metric < threshold && metric > threshold1)
        boundary = B{k};
        plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2);
        text(boundary(1,2)-35,boundary(1,1)+13,metric_string,'Color','r',...
            'FontSize',14,'FontWeight','bold');
    end
    % Count the segmented region with metric between 0.45 and 0.60 as 2fruit
    if (metric < 0.60 && metric > 0.45)
        boundary = B{k};
        plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2);
        text(boundary(1,2)-35,boundary(1,1)+13,metric_string,'Color','r',...
            'FontSize',14,'FontWeight','bold');
        count= count+1;
    end
end

count=(count+(length(fruit_idx)));
    %Show the total Dragon fruit found
    xlabel(['Number of Dragon fruit found:',num2str(count)]);

```

(e) Matlab coding for automated daisy counting (R-B)

```

[File_Name, Path_Name] = imgetfile(); %read the image user selected
flower= imread(File_Name); % assign the image to 'flower'
imshow(flower), title('Input Image'); %Show the input image
Blue=flower(:,:,3); %Blue Channel
Red=flower(:,:,1); %Red Channel
object=imsubtract(Red,Blue); %Red minus Blue channel
f=fspecial('gaussian', 3,3); %Define 3x3 gaussian filter
Filtered=imfilter(object, f); %Filter the image
r =1;g=2;b=3;
%Convert into binary image with threshold of 0.5
bw1=im2bw(Filtered, 0.5);
imshow(Filtered), title('Redchrominance Image');
Holescleared = imfill(bw1,'holes'); %Remove holes in the binary image
mask=strel('disk',2); % Disk shape mask with size of 3

```

```

Image_erosion=imerode(Holescleared,mask); %Erode image with diamond mask
figure,
imshow(Image_erosion);
mask2=strel('disk',3); % Disk shape mask with size of 3
mask1=strel('disk',2); % Disk shape mask with size of 3
Opening = imopen(Image_erosion,mask1);% Opening of image with square mask
Closing = imclose(Opening,mask2); %Closing of image with diamond mask
remove= bwareaopen(Closing,10); %Remove small object in image
imshow(remove), title('Binary Image of Segmented flower');

[L,n] = bwlabel(remove);
% to count the area of the region
flower_region = regionprops(L,'Area','BoundingBox');
flower_area = [flower_region.Area];
% find the regions that greater than 1% of total segmented region
flower_idx = find(flower_area>(0.01)*max(flower_area));
flower_shown = ismember(L,flower_idx);

for n = 1:length(flower_idx) % Create a outline of the region
    idx =flower_idx(n);
    flower_region(idx).BoundingBox;
    xmin = round(flower_region(idx).BoundingBox(1));
    ymin = round(flower_region(idx).BoundingBox(2));
    xmax = round(flower_region(idx).BoundingBox(1)) +
flower_region(idx).BoundingBox(3);
    ymax = round(flower_region(idx).BoundingBox(2)) +
flower_region(idx).BoundingBox(4);
end

%find the centroid of flower
S = regionprops(Image_erosion , 'Centroid');
centroids = cat( 1, S.Centroid );
figure,%Show the flower
imshow (flower), title('Detection Result'),....
    %Show the total flower found
hold on
    xlabel(['Number of flower found:',num2str(length(flower_idx))]);
    %Plot the centroid with blue marker size of 10
plot (centroids(:,1), centroids(:,2),'b+', 'MarkerSize',5);
hold off

```

(f) Matlab coding for automated daisy counting (Cr)

```

[File_Name, Path_Name] = imgetfile(); %read the image user selected
flower= imread(File_Name); % assign the image to 'flower'
imshow(flower), title('Input Image'); %Show the input image
k= rgb2ycbcr(flower); %Convert RGB to ycbcr
Redchrominance=k(:, :,3); %Shown Cr
f=fspecial('gaussian', 3,3); %Define 3x3 gaussian filter

```

```

Filtered=imfilter(Redchrominance, f); %Filter the Cr image
%Convert into binary image with threshold of 0.6
bw1=im2bw(Filtered, 0.6);
imshow(Filtered), title('Redchrominance Image');
Holescleared = imfill(bw1,'holes'); %Remove holes in the binary image
mask=strel('disk',2); % Disk shape mask with size of 3
Image_erosion=imerode(Holescleared,mask); %Erode image with diamond mask
figure,
imshow(Image_erosion);
mask2=strel('disk',3); % Disk shape mask with size of 3
mask1=strel('disk',2); % Disk shape mask with size of 2
Opening = imopen(Image_erosion,mask1);% Opening of image with square mask
Closing = imclose(Opening,mask2); %Closing of image with diamond mask
remove= bwareaopen(Closing,10); %Remove small object in image
imshow(remove), title('Binary Image of Segmented flower');

[L,n] = bwlabel(remove);
% to count the area of the region
flower_region = regionprops(L,'Area','BoundingBox');
flower_area = [flower_region.Area];
% find the regions that greater than 1% of total segmented region
flower_idx = find(flower_area>(0.01)*max(flower_area));
flower_shown = ismember(L,flower_idx);

for n = 1:length(flower_idx) % Create a outline of the region
    idx =flower_idx(n);
    flower_region(idx).BoundingBox;
    xmin = round(flower_region(idx).BoundingBox(1));
    ymin = round(flower_region(idx).BoundingBox(2));
    xmax      =      round(flower_region(idx).BoundingBox(1)      +
flower_region(idx).BoundingBox(3));
    ymax      =      round(flower_region(idx).BoundingBox(2)      +
flower_region(idx).BoundingBox(4));
end

%find the centroid of flower
S = regionprops(Image_erosion , 'Centroid');
centroids = cat( 1, S.Centroid );
figure,%Show the flower
imshow (flower), title('Detection Result'),....
    %Show the total flower found
hold on
    xlabel(['Number of flower found:',num2str(length(flower_idx))]);
    %Plot the centroid with blue marker size of 10
plot (centroids(:,1),centroids(:,2),'b+', 'MarkerSize',5);
hold off

```