DECISION SUPPORT SYSTEM FOR IN SITU MELON'S FRUIT HARVESTING TIME BASED ON FUZZY LOGIC AND SINGLE SHOT DETECTOR (SSD)

JAAFAR MOHAMMED JABBAR

COMPUTER SCIENCE
GRADUATE SCHOOL
IPB UNIVERSITY
BOGOR
2021

STATEMENT CONCERNING THESIS AND SOURCES OF INFORMATION AND THE EXTRAORDINARY OF COPYRIGHT

I declare that the thesis entitled "Decision Support System for In Situ Melon's Fruit Harvesting Time Based on Fuzzy Logic and Single Shot Detector (DSS)" is my work with the guidance of the supervisory committee and has not been submitted in any form to any college. The information sources derived or quoted from published or unpublished works of other authors have been mentioned in the text and are listed in the Bibliography at the end of this thesis.

At this moment, I assign the copyright of my writing to the IPB University.

Bogor, July 2021

Jaafar Mohammed Jabbar
G651198021

SUMMARY

JAAFAR MOHAMMED JABBAR. Decision Support System for In Situ Melon's Fruit Harvesting Time Based on Fuzzy Logic and Single Shot Detector (DSS). Supervised by SRI WAHJUNI and WILLY BAYUARDI SUWARNO.

Decision Support Systems (DSS) are wildly used in agriculture, industry, and healthcare, due to their efficient information management and decision-making activities. Melon has a considerably large genetic variability and is widely used in scientific research such as biology and genetics. In this study, we aimed to develop a system to detect the right harvesting time for melon fruits because the quality cannot be maintained after harvesting. Hence, it would be best if the harvesting is carried out at the right time, which will not affect the fruit harvested earlier. Secondly, melon fruit harvesting is done every day and not on the same day for all fruits due to environmental factors. The harvest DSS will classify melon's categories on the plant regarding melon's skin color by the guidance of the expert. The essential sensory factor for freshness and maturity is color, as mentioned in many studies. We use the skin color to classify melon fruit while it is on the plant into three categories: Ripe, About to Ripe, and Under Ripe, using a melon image. Firstly, localizing the melon on the plant using SSD and extracting its skin color. The input of fuzzy logic is the extracted color channels means values. The extraction of color is a challenging job because some fruits blocked with plant's stem, root, or leaves, so, we divided the process of reaching the purest melon skin into several steps: detecting the melon fruit on the plant, segmenting the detected melon and removing the unwanted part from the skin, and finally, applying image normalization and calculate mean values for melon's skin color. In this study, we achieved 91-100% accuracy in classifying melon ripeness levels. All images source were captured using iPhone rear camera.

Keywords: fuzzy logic, HSV color space, image processing, melon harvesting, SSD-Mobilenet.

DECISION SUPPORT SYSTEM FOR IN SITU MELON'S FRUIT HARVESTING TIME BASED ON FUZZY LOGIC AND SINGLE SHOT DETECTOR (SSD)

JAAFAR MOHAMMED JABBAR

Thesis submitted in partial fulfillment of the requirements for the award of
Master of Science degree in
Computer Science

COMPUTER SCIENCE
GRADUATE SCHOOL
IPB UNIVERSITY
BOGOR
2021

Thesis Examiners:

Dr. Medria Kusuma Dewi Hardhienata S.Komp.

digitally signed

disign.ipb.ac.id

Thesis Title : Decision Support System for In Situ Melon's Fruit Harvesting Time Based on Fuzzy Logic and Single Shot Detector (SSD)
Name : Jaafar Mohammed Jabbar
NIM : G651198021

Supervised by
Advisory Committee

Supervisor 1:
Dr. Ir. Sri Wahjuni M.T.

Supervisor 2:
Dr. Willy Bayuardi Suwarno S.P., M.Si.

Approved by

Head of the Study Program:
Prof. Dr. Imas Sukaesih Sitanggang, S.Si., M.Kom.
NIP 19750130 199802 2 001

Dean of the Graduate School
Prof. Dr. Ir. Anas Miftah Fauzi, MEng
NIP 196004191985031002

Digitally signed by:
Anas Miftah Fauzi
Date: 26 Jul 2021 16.41.42 WIB
Verify at disign.ipb.ac.id

Examination Date:
(25 Jun 2021)

Graduation Date:
()

FOREWORD

I would like to thanks almighty Allah because, without his permission, I would never have the ability to accomplish this thesis. I ask him to make all the research processes easy for me and make this research very beneficial and valuable for all human beings worldwide. I also express a deep appreciation to my family for their support in all of my endeavors.

Special thanks to my Supervisors, Dr. Sri Wahjuni and Dr. Willy Bayuardi Suwarno, for their advice, patience, guidance, and support. Thank you very much.

I appreciate the management of ATP Cikarawang, Mr. Sarwono, and Mr. Hendri Apriyanto in Jatiasih-Bekasi Melon farm to collect the data and provide melon information. Without your support, this feat would not be a reality.

Lastly, I would like to acknowledge the effort and contributions of all the staff at the Computer Science department, especially Dr. Mustafa. Thank you for all your support.

Bogor, July 2021

Jaafar Mohammed Jabbar

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

I INTRODUCTION

1.1 Background

Agricultural development is one of the most powerful approaches for ending extreme poverty, promoting shared prosperity, and feeding an estimated 9.7 billion people by 2050 (WBG 2020). According to World Bank Group (WBG), growth in the agricultural sector is two to four times more effective in increasing income among the poor than in other sectors. Employment in agriculture in 2016 analyses found that 65% of poor working adults earn their livelihood through agriculture. Agriculture is also essential for economic growth. In 2014, it accounted for a third of global GDP (WBG 2020). In Indonesia, agriculture plays a leading role in promoting food security. It is also closely related to employment, wages, and local economies because most people in Indonesia depend on farm products (Panuju et al. 2013).

The agriculture development using a decision-making system such as the Decision Support System (DSS) is encouraged by several retailers, practitioners, and academics because DSS activities have created high expectations and optimism on the prospects of better decision-making (Power 2002). Agricultural nutrition and other integrated pesticide management (IPM) are widely accepted with DSS applications (Bange *et al.* 2004).

It is challenging to apply the Decision Support System (DSS) in agriculture due to the farmer and scientists' information management. On the other hand, DSS help in increasing the economic and crop productivity rate. It is essential to explore, and model data flow between decision-making processes and the user's feedback for successful outcomes. And this can be done by implementing the DSS, which provides the selection of crops with precise and comprehensive agricultural details (Venkatalakshmi and Devi 2014).

"Melon has inner and outer characters, including skin color, size, aroma, and taste can determine the melon quality." These must be preserved before harvest, as production cannot be maintained after harvest. The fruit quality is based chiefly on plant genetics, critical fruit production, managing, and maturation (Coelho *et al.* 2007; Freilich *et al.* 2015). The essential sensory factor is color because it is strongly correlated with qualitative factors like freshness and maturity (Kays 1999). Based on that, transferring the human experience to the machine to decide whether the fruit is ripe will reduce human activities and reduce the production cost. Several studies applied machine learning techniques to simulate the human experience, such as detecting plant disease (Chen *et al.* 2021), classifying the ripeness level for apple fruit (Dadwal and Banga 2012). Many object detections models have been developed using a deep learning algorithm, such as Faster R-CNN and SSD-Mobilenet. Faster R-CNN has a higher detection accuracy but lower processing time than SSD-Mobilenet (Galvez et al. 2019; Ghoury et al. 2019), where time and memory consumption are essential factors in limited computing capabilities devices such as Raspberry Pi. The publication of Ghoury et al. (2019) was conducting a pre-trained mobilenet model with SSD to identify grape and classify if it's normal or has a disease using transfer learning. They have used a total of 656 images, including Healthy Grape, Diseased Grape, Healthy Grape Leaf, Diseased Grape leaf labelled with the LabelImg application (Tzutalin 2015). The training was 6981 steps for Faster R-CNN and 16229 steps for SSD-Mobilenet. The study states that the SSD works faster than the Faster R-CNN, but less accurate with small objects.

In return to melon fruit, harvesting is a daily task because ripening is uneven simultaneously. Therefore, the farmers' profits will be reduced if collectors conduct total soluble solids (TSS) inspections to define the maturity level. Determining the maturity of a fruit by humans is a dynamic phenomenon based on internal and external factors, as discussed in Ahmad (2017). The selection of an object maintained by a person's eyes regarding the person's experience will increase the production cost and time consuming. On the other hand, the automated decision that doesn't possess the human mind uses a sensor, camera, or another data source instead. The automated decision maintains the selection process and provides crop information to the decision maker, so that the human activities will be reduced. The automated decision processes the information of the fruit that comes from a camera or other data source by applying machine learning techniques to recognize the target fruit and correctly localize it (Zheng *et al.* 2015; Zheng *et al.* 2016).

The melon used in this study is described in Iskandar *et al.* (2019). It is honeydew *(Cucumis melo* L.*)* of Alisha F1 cultivar, that has a flattened shape and a crunchy flesh texture. Its under-ripe skin color is green, and the ripe one is yellow. The flesh color is a white-orange. The harvesting decision is based on the fruit's skin color. Thus, an image segmentation technique has been used to extract the skin color, and a fuzzy logic system to classify the ripeness levels into ripe, about to ripe, and under-ripe categories, to find the desired harvest time.

Fuzzy logic establishes genetic algorithms which imitate a section of human reasoning. The methods are synthesized by establishing a computer program called a fuzzy rule-based system (Amendola *et al.* 2005). The theory that applied mathematics to diffuse concepts was proposed by Lofty Asker Zadeh in 1965 and tried to approach human reasoning by fuzzy sets, also described by linguistic variables (Zadeh 1997).

Fuzzy logic has a statistical output for which can be used for better and accurate decision-making in such a variety of applications. The fuzzy control system is represented in shape close to the primary language form for showing the actual knowledge needed for a task in different manners. Fuzzy logic does not need to be modeled using a complex mathematical design and easily converting the expert experience to rules programmatically. Finally, system behavior can be implemented and adapted easily and quickly. As a result, the long-term modification will quickly be done (Zadeh 1997; Wang and Qiu 2003). Using this kind of logic system to classify fruit ripeness will produce an accurate information about maturity levels, therefore the harvesting and selection of the fruit will be managed easily by the decision maker or user through a Graphical User Interface (GUI).

1.2 Problem Statement

1. Melon harvesting tasks are costly due to the melon's inconsistent ripeness time, so harvesting is done every day and not on the same day. In view of this, more workers are needed for harvesting.
2. The light environment varies between fruits regarding the position of the fruit, e.g., if the melon was in the middle, the lightness will be deferred from melons in the corner.

3. The freshness cannot be managed after harvesting, therefore, it's crucial to detect and classify melon ripeness levels on the plant.

1.3 Research Goals

This research aimed to:

1. Detect melon fruits on the plant.
2. Provide information about the ripeness of melon fruit to determine if it is ready to be harvested.

1.4 Research Benefits

The benefit of this research is to help the farmer and scientist with organized information to efficiently manage the harvesting remotely, using the camera of a harvesting robot. This will reduce the cost of production for melon fruit by reducing human activities.

1.5 Research Scope

The research is primarily concerned with detecting melon fruit and classifying the ripeness level to find the exact harvesting time based on melon's color. The data are collected from a farming greenhouse in Agribusiness and Technology Park (ATP) IPB Cikarawang. The melon used in this study is honeydew *(Cucumis melo* L.*)* of Alisha F1

II LITERATURE REVIEW

2.1 Decision Support System

A Decision Support System (DSS) is a computer software technology that evaluates and displays business information so that consumers could more effectively make business decisions. It is an informative framework (to differentiate it from an operational application that gathers data during regular business operations) (Power 2002; Kopáčková and Škrobáčková 2006), specific information that may be collected and provided by a decision support system are:

- Comparative sales over the week to the next.
- Projected income basing on existing market expectations for goods.
- The implications of various possibilities to decisions provided previous experiences in a described context.

A decision support system can graphically display data and have an expert system or artificial intelligence (AI). The DSS could target business leaders or any other category of information staff.

Many researchers have defined and used DSS. As in Hartati and Sitanggang (2010), in modern farming, evaluating land suitability and selecting crops is of the greatest priority to any enterprise because the smaller piece of the land demands more flexibility in farming in line with the specifications region. The selection criteria for the correct soil type and selection of plants as per decision-makers are dynamic and unstructured. Hence, their fuzzy-based DSS can depict and exploit agriculture information that is imperfect or ambiguous and can be used to assess land constraint scores.

2.2 Single Shot Detector

SSD is the state-of-the-art object detector network introduced by Liu *et al.* (2016). The mAP (mean Average Precision) of 74% for 59 frames per second on *Pascal-VOC* and *COCO* datasets. The dataset consists of 328,000 images and 2.5 million labels, and it has been used to perform a deep-learning object detection model training, known as COCO (Common Objects in Context) dataset obtained by Lin *et al.* (2014).

Single-shot refers to a single forward pass of the network for localizing the object and classified it. Multi-Box refers to the bounding box regression method used for fast bounding box coordinate proposals. Detector refers to the detection network that classifies the detected objects also. The SSD network architecture was built on the VGG-16 architecture (stands for Oxford Visual Geometry Group-16 layers), as shown in Figure 1, by (Simonyan and Zisserman 2015), and without the *fully-connected* layers due to the performance of VGG-16 in classifying images with high qualities. Also, the reason for *transfer-learning* that helps with results improvement. They added a supporting layer to extract multi-scale features and reduce the subsequent layers' input size. When finishing the extraction of features by the convolution layers, we get a feature layer of $M \times N$ size (i.e. the number of locations) with p channels, same as in Figure 2 (b and c) 8×8 and 4×4 respectively then applying a 3×3 convolution layer (conv) to feature layer

$M \times N \times p$. From these locations we will get a k bounding boxes. The k bounding boxes have a different aspect ratio and size. Then computing c *class scores* for each box and 4 offsets same as the original box shape.

Figure 3 shows a 1×1 convolution that helps dimensionality reduction where the number of dimensions will reduce and keep the width and height without any changes.

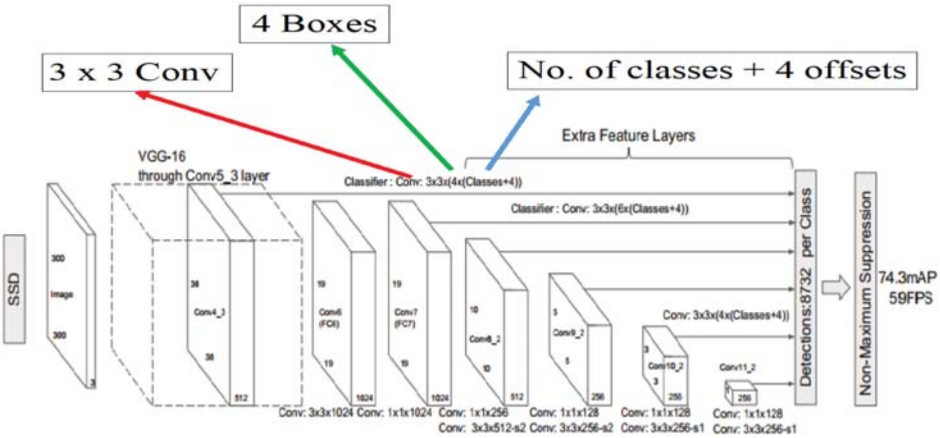
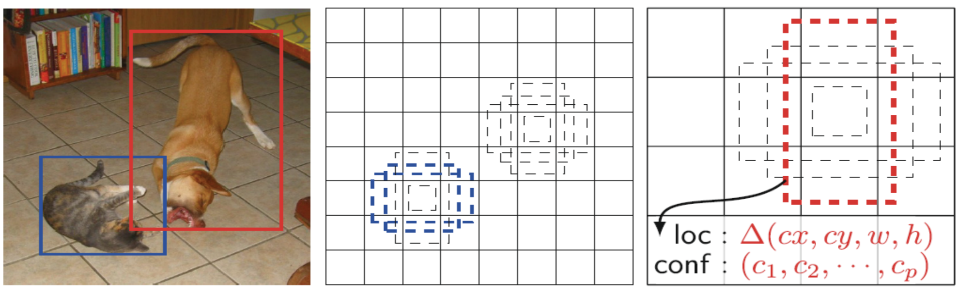


Figure 1 Architecture of Single Shot Multi-Box detector with the input of 300x300x3 by Liu et al. (2016)



(a) (b) (c)

Figure 2 SSD Framework. (a) GT boxes, (b) 8 × 8 feature maps, and (c) 4 × 4 feature maps by Liu *et al.* (2016)

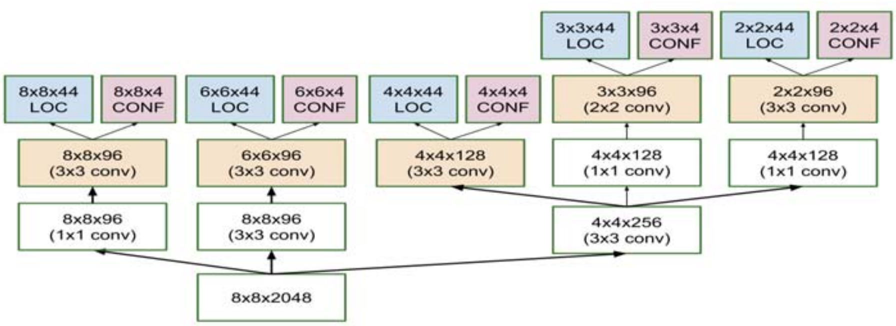


Figure 3 A prediction of multi-scale convolutional for localization and confidence used for SSD in (Liu *et al.* 2016) study

2.2.1 Loss Function

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)) \quad (1)$$

Where,
N : number of matched default boxes,
Alpha (α) : scaling factor for localization loss,
l : predicted bounding box,
g : ground truth bounding box,
c : softmax activated class score for default box i with category p
x : matching indicator between default box and ground truth box j

Equation (1) is the loss function of the multi-box consists of confidence loss (L_{conf}) .And the location loss (L_{loc}) (i.e., loss between the predicted box and the Ground-truth (GT) box. N Is the equated default boxes. The alpha is to make a balanced contribution of the L_{loc}. We know that we want to achieve the most reduced loss function values in deep learning, so prediction becomes equal or partial equal to the ground-truth box coordinates. Multi-Box doesn't work on object classification, whereas SSD performs an object classification. So, for each bounding box prediction, a set of class predictions are calculated for each class.

2.2.2 Scales and Aspect Ratios of Default Boxes

The scale of default boxes computed as follow in Equation (2),

$$S_k = S_{min} + \frac{S_{max} - S_{min}}{m-1}(k-1), k \in [1, m] \quad (2)$$

If we have a prediction of m feature maps, then we calculate the S_k at k_{th} feature map. S_{min} It is 0.2 and S_{max} is 0.9. The scale is meant to be at 0.2, the lowest layer, whereas 0.9 is on the highest. At each, S_k five aspect ratios have been defined for non-square bounding boxes, as shown in Equation (3) below,

$$a, \in \left\{1,2,3,\frac{1}{2},\frac{1}{3}\right\}, (w_k^a = S_k\sqrt{a_r})(h_k^a = \frac{S_k}{\sqrt{a_r}} \quad (3)$$

For the 1:1 aspect ratio, we got 1 square bounding box, Equation (4),

$$S'_k = \sqrt{S_k S_k + 1} \quad (4)$$

Thus, we got a six bounding box that has different aspect ratios.

2.2.3 Multi-Box and Intersection over Union (IoU)

To evaluate an object detection model and see how accurately a bounding box is predicted, we can use the Intersection over Union (IoU) (Rezatofighi *et al.* 2019), to compute how much the predicted bonding box similar to the ground-truth (GT) bounding box. The IoU measures the distance between two shapes, and that is also known as the *Jaccard index*. The good detection should have IoU greater than 0.5, as you can see in the image below in Figure 4 a 0.5 IoU computed by Equation (5). It still not a good one, but it's better than if we compared it with a random bounding box coordinates hence, multi-box will start regressing from a known coordinate that is near from GT bounding box.

$$IOU = \frac{|A \cap B|}{|A \cup B|}, where \ \frac{|A \cap B| \text{ is Area of Overlap}}{|A \cup B| \text{ is Area of Union}} \tag{5}$$

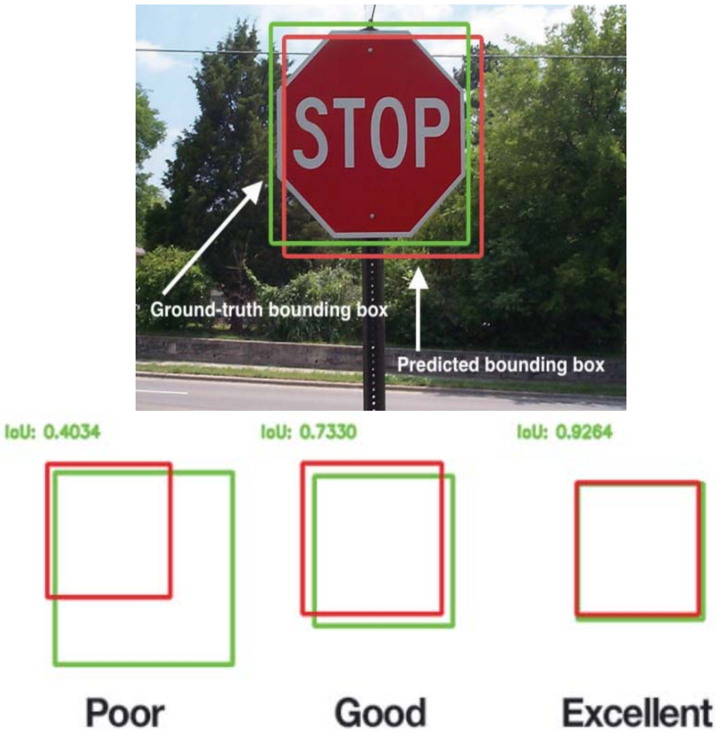


Figure 4 An explanation for IoU

2.3 Image Segmentation Based HSV Color Space

Image segmentation is a method that segments an image based on specific features in the image. For segmenting an RGB image (i.e., colored image, *r* for red, *g* for green, and *b* for blue channels), we can use HSV color space. The HSV color space is used on colored image segmentation to find the color range with the Region of Interest (ROI). It is a cylindrical color representation, whereas RGB color space a cubic color representation. RGB color space is called the pixel depth that is $M \times N \times 3$ array of pixels. A 3D in RGB color space pixel $p(i)$ is defined by red, green and blue at pixel coordinates $(r(i), g(i), b(i))$, implies for three columns (R for red, G for green and B for blue channels) (Kumar and Verma 2010; Marques 2011a). There are many types of color space, and the most used are LAB and HSV (Sural et al. 2002; Bora and Gupta 2014). According to Bora et al. (2015) experiment that compares LAB with HSV, they have found that the HSV color space is the best choice for colored image segmentation.

HSV stands for (Hue Saturation Value), Hue (H) is described as the range of (0, 2pi) for red axis at 0 degrees, green at 2pi/3, blue at 4pi/3, and red at 2pi. Saturation (S) is used to represent the pureness of hue with white as a reference. It has been expressed as the depth of color and computed as a radial stretch of axis center between 0 to 1(Sural et al. 2002; Zhao et al. 2002). When saturation is 0, one rises to the intensity axis with whiteness increasing. If the saturation went from 0 to 1, the color would become purer by its hue (Sural et al. 2002).

Value (V) is the brightness of the HSV color space. The V factor represents the lightness/brightness in the image color. The V range is 0 to 100 (Bora et al. 2015). E.g., the brightness will become more if the hue is red with high value and darker with low value (Bora et al. 2015). Figure 5 shows the HSV Color Space representation (Wikimedia).

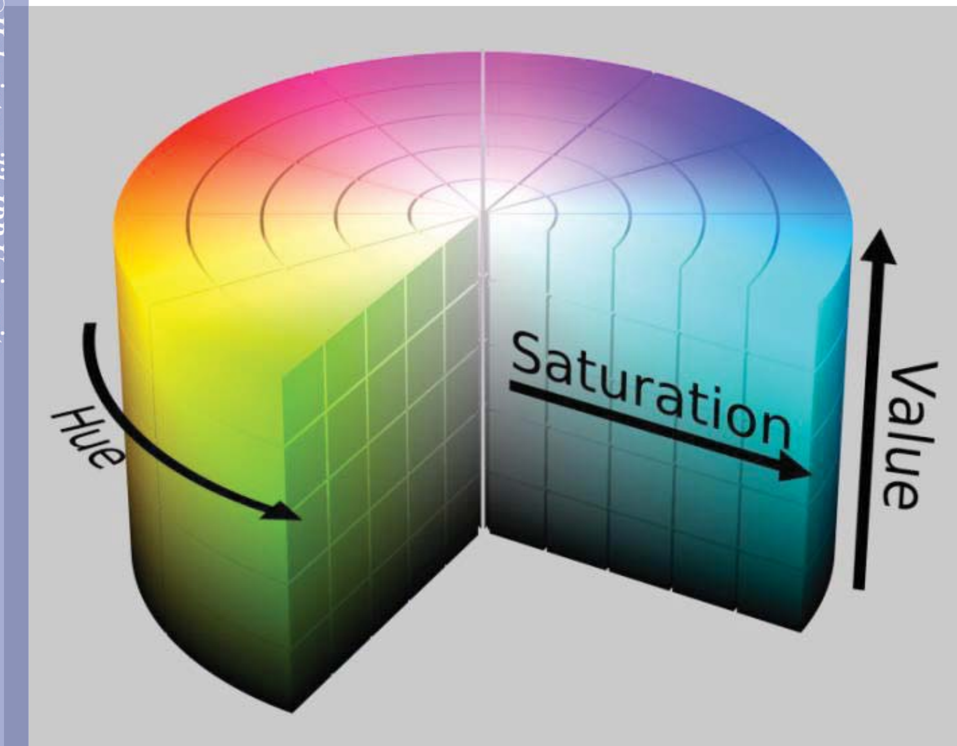


Figure 5 HSV Color Space representation

RGB normalization is an important factor in terms of color differences to keep the pixel intensities in a standard range. For example, at 11 AM, the RGB values of a pixel in an image are (200, 150, 120), and at 4 PM are (190, 170, 100). So we have to normalize these values to get rid of lightness changes. The typical approach to normalize RGB values is MIN-MAX standardization (Marques 2011) by using Equation (6) to (11).

$$f(x,y) = (r,g,b) \tag{6}$$

$$Total = r + g + b \tag{7}$$

$$R' = \frac{r}{Total} \times 255 \tag{8}$$

$$G' = \frac{g}{Total} \times 255 \tag{9}$$

$$B' = \frac{b}{Total} \times 255 \tag{10}$$

$$g(x,y) = (R',G',B') \tag{11}$$

2.4 Fuzzy Logic

Fuzzy logic has several computation types that are used regarding the number of input and output. The commonly used one is Mamdani for this kind of work that needs to translate the expert's natural language to the machine by formulating what he suggests as IF-THEN rules.

Physicalisation is the way of adding machine numerical data to a fuzzy system from a particular membership grade. Such membership degree will occur anywhere inside the (0 to 1) range. If this is zero, then the number is not connected to the given fuzzy collection, but if it's one, then the number is in the fuzzy set entirely. Any value within 0 and 1 is the level of ambiguity found in the set rate. Typically, such vague sets are represented by words, and likewise, by applying machine information to fuzzy sets, we can naturally argue from it linguistically (Zadeh 1997; Pedrycz et al. 2016). For example, Figure 6 shows the structure of a device for monitoring the temperature operated by a fuzzy logic controller. 62 and 78, then it belongs to fuzzy set with membership grade 0.10 and 0.90 respectively (i.e., the first member is not more likely belong to the HOT set, but 78 temperature is most likely belong to it).

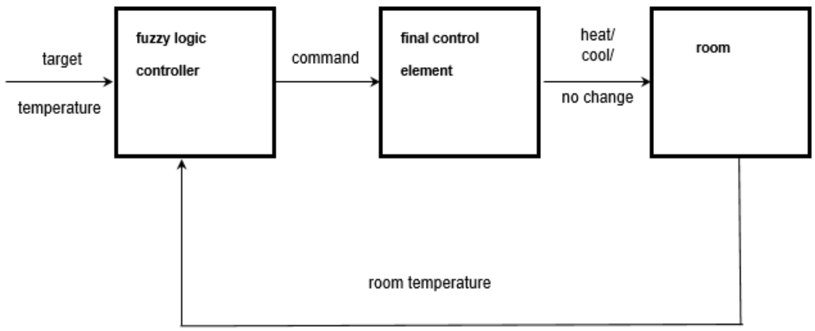


Figure 6 A simple fuzzy logic system to control room temperature by Singhala et al. (2014)

Illustrate the use of a fuzzy logic system. Data such as room actual temperature and desired value by the specified method can alter the ambient temperature. The relation between the ambient temperature and the desired temperature can be contrasted with the fuzzy engine after a certain time interval and generates heating or cooling order Singhala et al. (2014). The membership classification is correlated with it in a fuzzy set community. For e.g., the collection of HOT temperature within 60 < TEMP < 80 is defined. In a fuzzy set, the member has their membership grade associated with it. For example, the set of HOT temperature is decided between 60< TEMP< 80. If the mercury is 60 degrees, we conclude that it does not refer to the HOT group, but it belongs to the fuzzy logic system, which has a rating of 0. Likewise, suppose the mercury is 62 and 78. In that case, it belongs accordingly to the fuzzy group of membership category 0.10 and 0.90, i.e., the first member is not much likely to belong to the HOT group, but the group of HOT is still most apt to have 78 temperature. Figure 7 below shows the membership function.

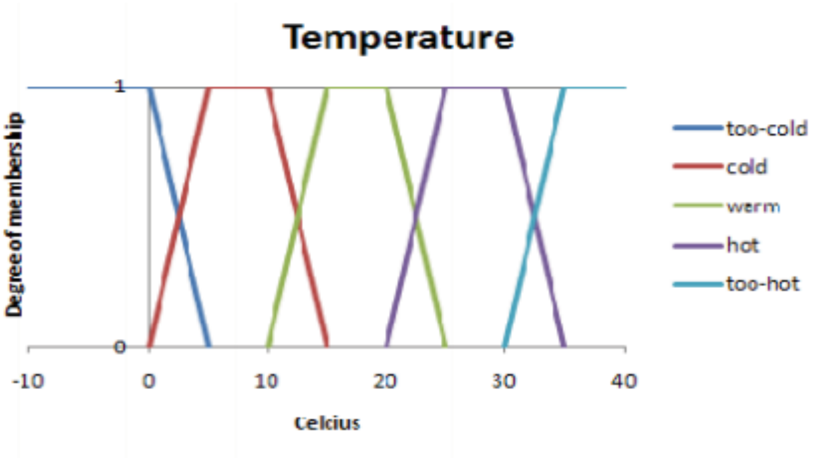


Figure 7 Membership functions for T (temperature) = {too-cold, cold, warm, hot, too-hot} (Source: Singhala et al. 2014)

According to Li et al. (2019), implementing an irrigation decision support system (IDSS) offers sensible advice for effective agriculture performance. Thus, the implementation of DSS serves an essential role in enhancing irrigation automation and irrigation practices on Alfalfa farms. DSS architectures were centered on a fuzzy control method and implemented to design alfalfa growth and the model of soil water to promote irrigation scheduling. The machine can depend on soil water and weather forecasts to decide the phase of development of alfalfa. The fuzzy control method calculates the rate and volume of irrigation in real-time by pointing at the soil's water and the clover's varying elevation. The application has been created using Java-Eclipse, featuring a user-friendly GUI, and simple for peasants with basic knowledge of irrigation scheduling and IDSS technical expertise.

A series of optimization standards is undertaken for decision-making according to planning criteria. The current assay develops a DSS to detect melon in the image using a Tensorflow pre-trained deep learning model SSD (Abadi et al. 2016), then employing Fuzzy Inference System (FIS) to manage melon's fruit aiming at good cultivation. In the study of (Dadwal and Banga 2012), they developed a fuzzy logic system to classify the ripeness levels of apple fruits into (ripe, under ripe and over ripe categories) based on RGB color space.

As we mentioned above, the colored image in RGB color space consists of three layers (*r* for red channel, *g* for green channel, and *b* for blue channels) that will produce the corresponding color of each pixel. Thus, they have calculated these channels' means values (i.e., *r, g*, and *b*) and using them as fuzzy logic inputs.

They have captured an apple fruit from 4 sides (i.e., four images for one apple fruit). Figure 8 shows one of the apple fruit sides in (a), and (b) is the same image after color segmentation. They have analyzed 80 images of apple to define reference values for classifying the ripeness based on *r, g*, and *b* mean values to build their fuzzy rules.

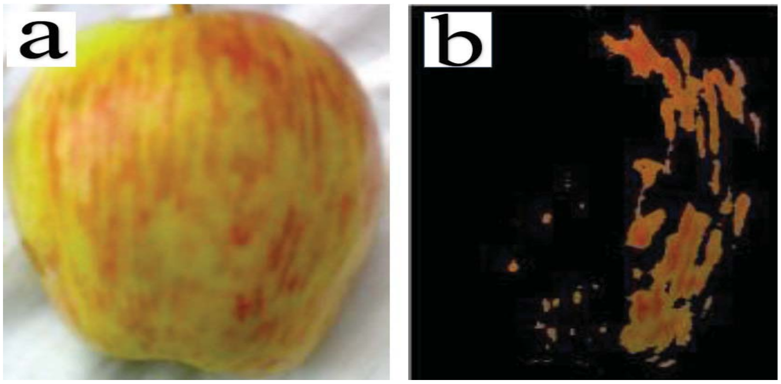


Figure 8 a) Is the apple fruit used to develop a fuzzy logic system to classify apple fruit ripeness levels, and b) is the apple's image after segmentation (Source: Dadwal and Banga 2012)

The process of their system as follows:

1. They used the image segmentation technique to find the area with the red color of the corresponding image, as shown in Figure 8 (a and b) above.
2. Extract the *r, g*, and *b* channel values from the segmented image and calculate each channel's mean value.
3. Using the mean values of the *r, g*, and *b* channels as inputs for the Fuzzy Logic system, the membership functions for *r, g*, and *b* are built using *MATLAB* software, as shown in Figure 9 (a to f).

The reference values are based on 80 images to build the fuzzy rules. That will fires when the input image of apple fruit satisfies the rule. For example,

1. $r(i) > a$: *means that the primary colour component* (*red*) *should be larger than a*.

2. $g(i) < b$: *means that the primary colour component* (*green*) *should be larger than b.*

3. $b(i) < c$: *means that the primary colour component* (*blue*) *should be larger than c.*

The following are a group of their fuzzy rules,

1. If (red is low) and (green is low) and (blue is low) then (category is under ripe).
2. If (red is high) and (green is high) and (blue is high) then (category is overripe).
3. If (red is medium) and (green is medium) and (blue is medium) then (category is ripe).
4. If (red is very high) and (green is very high) and (blue is very high) then (category is overripe).

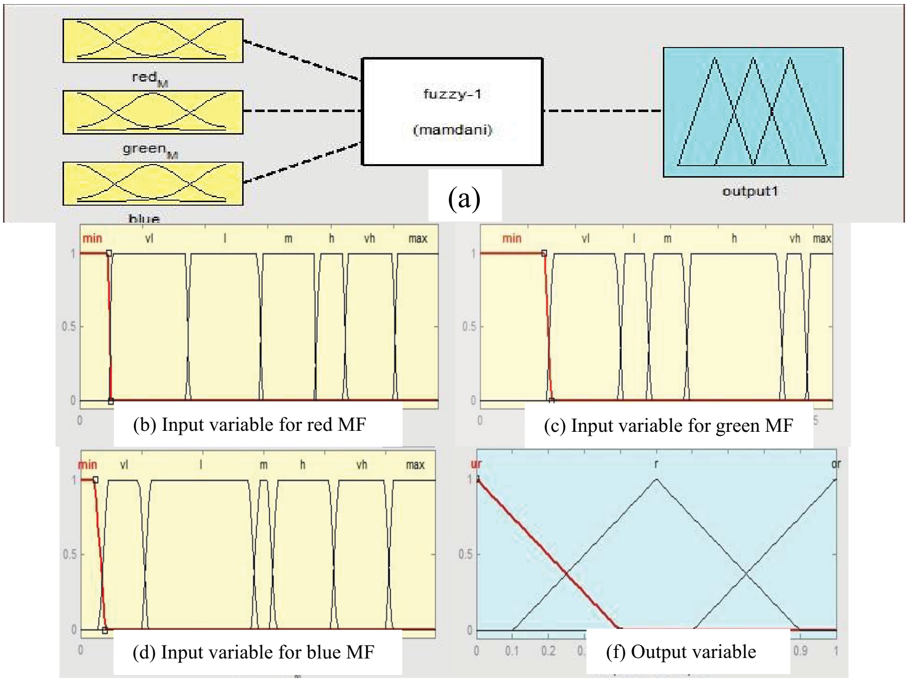


Figure 9 Dadwal and Banga (2012) fuzzy logic system process for classifying apple fruit ripeness levels, developed by (Dadwal and Banga 2012) using *MATLAB* software. In (a) is the system structure, there are three membership functions (MFs) inputs and one output. (b) The *r* MF input, (c) is the *g* MF input, and (d) is the *b* MF input. Finally, (f) is the Defuzzification output

The Defuzzification output explained as follow:

If Defuzzification output is less than 1, Category is under ripe (ur).

If Defuzzification output lies between 1 to 3, Category is about to ripe (atr).

If Defuzzification output lies between 3 to 5, Category is overripe.

If Defuzzification output lies between 5 to 7, Category is overripe.

The system can be applied to another fruit classification by defining new references regarding the exciting color.

The system also fails at some point, as they said because they built it based on images with prepared lightness environmental. Another camera or different lightness will be different pixel intensities and cannot match their reference values.

III METHODS

3.1 Data and Area Studies

The data used are images of honeydew *Cucumis melo* L. cv. Alisha F1 melon. Image acquisition is made using a rear iPhone 6S camera, and default image size 4032 × 3024 pixels). The images were captured at *Agribusiness and Technology Park* (ATP) Cikarawang greenhouse owned by IPB University in Bogor, Indonesia. When capturing, the camera distance was 25-35 cm with an eye-level angle about the same level as the melon fruit. It was captured in July 2020, the time was 4 PM to 7 PM. The initial condition of the melon age was 60 days at the first attempt, and we continue capturing every two days until the age of 78. The harvesting was every day. Thus, at age 78, the last harvesting was done of the last residual melon group. The planting of all melons in the greenhouse was on 15/May/2020.

A new 50 images were captured for available melons categories of new planting in the same greenhouse on 20/February/2021. The shoot was taken using iPhone XS camera with the same distance, direction and image size as mentioned earlier.

The total images are 190, where 170 are the captured images, and 20 images were collected from Google-Image to increase the negative class. The images collected from Google-image are (unwanted objects such as background, leaves, and root), it is a common way in training an object detection model in order of increasing the detection accuracy. All images were resized to 640 × 480 pixels, and the labeling of the images was done by the greenhouse expert's guidance.

3.2 Equipment

The equipment we used in this research is explained as follow:

3.2.1 Hardware:

a) iPhone 6S, rear camera: 12 MP (f/2.2, 29mm, 1/3 inch), and default image size 4032 × 3024).
b) iPhone XS, rear camera: 12-megapixel (f/1.8, 1.4-micron) + 12-megapixel (f/2.4), and default image size 4032 × 3024)

3.2.2 Software:

a) A Windows 8.1 running on Core i7 processor, 8 GB RAM.
b) Python 3.7 Programming language.
c) Tensorflow 1.15 Object detection API.
d) Sickit-fuzzy Library.
e) Google Colab (Google) with Nvidia Tesla V100 SXM2 32 GB GPU and 13 GB RAM.

3.3 Research Stages

The harvesting time varies for each fruit in the greenhouse. There are ripe and under-ripe melons at the same age. The greenhouse expert decided whether the melon fruit is ripe and ready to be harvested or not, depending on the fruit's skin color. Thus, we need to convert this experience from the expert/farmer to the machine. To find the ripeness level, we have to dive into four steps, as shown in Figure 10, and will be explained in detail in the following methodology.

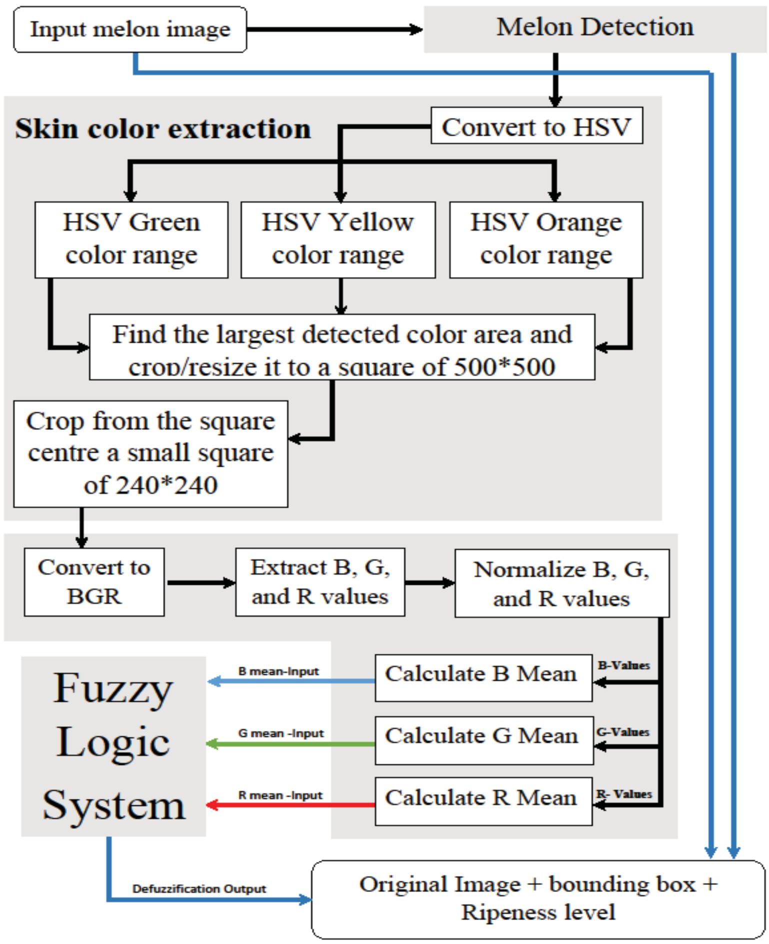


Figure 10 The process of the harvesting time Decision Support System. 1) Detect the Melon by SSD-Mobilenet, 2) extract skin color, 3) convert to BGR and prepare fuzzy logic inputs, and 4) use *b, g*, and *r* mean values as fuzzy logic system inputs and calculate Defuzzification output

3.4 System Development

3.4.1 Melon's Fruit Detection

The first step of this method is to find the extensive melon fruit in the image. It is noted that there is a possibility of multiple melon fruit existed in one image due to the environmental condition.

Based on the above, the dataset is separated into two classes. Our desired Melon class and the other class are the backgrounds that included images for unwanted objects (e.g., small melon, far or on the left/right corner of the image, plant stem, leaves, and roots). The images are annotated by drawing a rectangle manually (i.e., the bounding box) on each object in the image. e.g., if the image includes the Region of Interest (ROI), then annotate it as melon, and annotate all other parts of the image as a background. The total annotated objects in 190 images are 554 annotations (554 bounding boxes coordinates for each object in all images). The annotation process is done by using the LabelImg application (Tzutalin 2015). The dataset was divided into 85% for training, and 15% for testing. The output of this step is the detection of the desired melon in the image as shown in Figure 11.

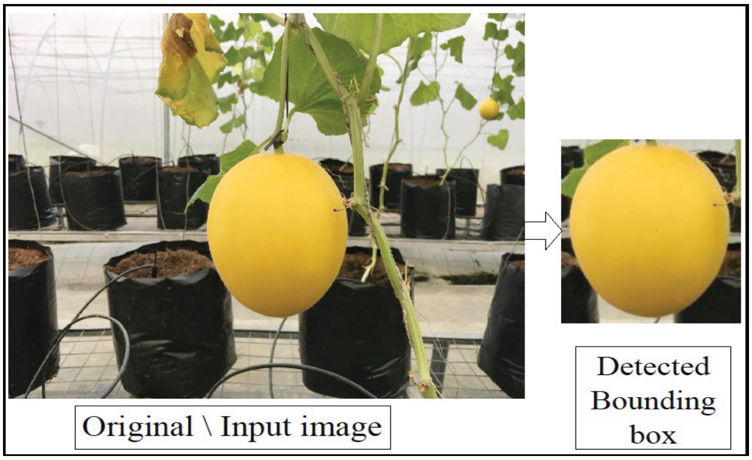


Figure 11 The detection process using SSD model

3.4.2 Image Segmentation

Color Image segmentation is to partition the image based on some features. In this step, the segmentation will be based on color. The input image used is colored in RGB color space pixel p(i), which is defined by red, green, and blue at pixel coordinates (r(i), (g(i), b(i)), implies for three columns (R for red channel, G for green channel, and B for blue channel) (Kumar and Verma 2010; Marques 2011b). The image segmentation was applied after detecting the desired melon and recognize it from the greenhouse environment, then derive to this step to extract skin color by segmenting the image to remove noise such as tee's root or a part of leaf blocking the fruit. Also, to be sure that we got a pure skin color extraction, to calculate the *r, g*, and *b* means of the skin color.

There are many types of color space, and the most used are LAB and HSV (Sural *et al.* 2002; Bora and Gupta 2014). In this (Bora *et al.* 2015) study, they have been used image segmentation to compares LAB with HSV, and they have found that the HSV (Hue Saturation Value) color space is the best choice for colored image segmentation.

3.4.3 Preparing Fuzzy Logic Input

Sun reflection or other light noise may change the pixels' intensities. Thus, the reference values that we will set later may or may not matches the future lightness effects on the inputted image and will lead to mismatched fuzzy rules with the fuzzy input (i.e., *r, g*, and *b* mean values). Undoubtedly, the classification between ripeness categories will be less accurate because of the reference values that settled regarding different lightness situations.

Accordingly, we need to normalize the pixel intensities for the skin sample produced in the previous step. As a result, *r, g*, and *b* will be in or close to the reference ranges of *r, g*, and *b*. Therefore, this step explained in two parts:

a) *Normalizing RGB*

RGB normalization is used to keep the pixel intensities in a standard range, using MIN-MAX standardization method. The maximum value in RGB channels is 255 representing a white pixel, and the minimum is 0 representing a black pixel. Equation (6) to (11) is used to normalize *r, g*, and *b* pixels intensities.

b) *RGB Mean Calculation*

The Mean of each channel of RGB color space of the sample area of the image is used as input of the fuzzy logic (i.e., the mean for each column in the image matrix). The Equation for obtaining the mean of each color layer in RGB color space is shown in (12), (13), and (14) below. This step was performed on a Melon image that is mapped from the previous step. The range value (minimum and maximum) of the RGB value for each category (Ripe, About to Ripe, and Under Ripe) is obtained from the above calculation. From this standpoint, we are using these range values as a reference to classify the melons into their categories.

$$R\mu_X = \mathrm{R}'/\text{No of Pixels} \qquad (12)$$

$$\mathrm{G}\mu_X = \mathrm{G}'/No \text{ of Pixels} \qquad (13)$$

$$\mathrm{B}\mu_X = \mathrm{B}'/No \text{ of Pixels} \qquad (14)$$

Where, R' = Normalized red pixels , G' = Normalized green pixels , and B' = Normalized blue pixels.

$\mathrm{R}\mu_X$ = Mean value of the normalized red layer,

$\mathrm{G}\mu_X$ = Mean value of the normalized green layer, and

$\mathrm{B}\mu_X$ = Mean value of the normalized blue layer.

3.4.4 Fuzzy Logic to Detect the Ripeness of Melon Fruit

We are using fuzzy logic to classify the melon fruits into ripe, about to ripe, and under-ripe categories. The classification is made based on the original images mentioned earlier in this paper, the images captured by the iPhone 6s, rear camera. The melon images are already labeled with their corresponding ripeness categories by expert guidance. The fuzzy logic algorithm is selected due to its ability to convert the expert's experience to computational logic. The fuzzy logic algorithm process is explained in three steps. Firstly, Defining the input/output Membership Function (MF). Secondly, setting the rules and combines all the rules and MFs in the control system. Lastly, producing the output for each rule based on the input.

The Membership Functions (MF) were created after exploring all the mean values of all images and selecting the reference values. We have created the MFs for red, green, and blue layers based on the reference values as shown in Figure (14). For example, the MF of the medium red layer (M_red) is a *Trapezoidal* function. And can be defined in Equation (15), (16), and (17) below (Zadeh 1997; Pedrycz *et al.* 2016).

$$\mu_{M\_red}(x) = \begin{cases} 0, & (x < a) or\ (x > d) \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \end{cases} \tag{15}$$

Where, $a < b < c < d$,

1. a = M_red Lower limt = 80
2. b = M_red Lower support limit = 95.
3. c = M_red Upper support limit = 105.
4. d = M_red Upper limit = 110.
5. x Is the input of the MF.

The other cases of trapezoidal function are R-function and L-function Figure (15) (a)Red (L_red and H_red MFs respectively).

R-function: $a = b = -\infty$ (L_red), Equation (16).

$$\mu_{L\_red}(x) = \begin{cases} 0, & x > d \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 1, & x < c \end{cases} \tag{16}$$

L-function: $c = d = +\infty$ (H_red), Equation (17).

$$\mu_{H\_red}(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & x > b \end{cases} \tag{17}$$

The computation of all MFs is done with the help of scikit − fuzzy library. scikit − fuzzy also known as skufzzy, is a fuzzy logic toolbox for Python programming language by (Warner et al. 2019). Figure (16) is the Defuzzification output.

IV RESULTS AND DISCUSSION

The classification of the fruit ripeness while the fruit on the plant is a difficult task. Lighting that varies from one to another fruit in the greenhouse is impacted by direct sunlight, shadows, or other light noise.

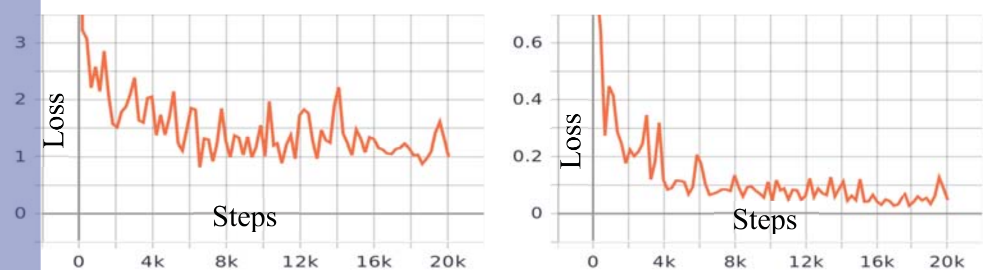
4.1 SSD Model Training

The model is trained on 554 bounding boxes (20% background, 20% melons not in the center of the image (negative class)), and (60% for the desired class, that is the most extensive melon object located in the middle of the image). A Pre-trained deep-learning model SSD-Mobilenet is preferred for extensive object detection in an image (Liu *et al.* 2016). The model was used with Tensorflow object detection API (Abadi *et al.* 2016) to train the images dataset on Colab pro. Also known as Google collaboratory, the Colab-pro version is to provide a consistent huge GPU by subscribing for a periodic payment (Google). The SSD- Mobilenet configuration for the training process is shown in Table 1, where the (Max detections per class and the Max total detections) is 2 for both, to force the detection in detecting the preferred object that is where the camera pointing and not the melons in the background, also controlling what class/object we want to detect easily.

Table 1 SSD-Mobilenet configuration

No	SSD Training parameters	Values
1	The input image size	480 width, 360 height
2	Batch size	24
3	Number of layers	6
4	Min scale	0.2
5	Max scale	0.95
6	Aspect Ratios	1.0 2.0 0.5 3.0 0.3333
7	Initial learning rate	0.004
8	Max detections per class	2
9	Max total detections	2

The training was 20,000 steps takes around 5 hours, with good low loss for localization and classification, as shown in Figure 12, where (a) graph is the Classification Loss of the hostile class objects and the extensive Melon class object, and (b) graph is the localization loss that is the best loss for the object detection task when using Tensorflow API. As we can see, the localization loss was the best at step 18,853, so we use this step because our interest is more toward the localization rather than the classification in this step.



(a) Classification Loss (b) Localization Loss

Figure 12 The losses of Tensorflow SSD-MobileNet training

After training has been accomplished, converting the trained model, which contains the model structure with the values of the required variables, such as weights to $OpenCV - dnn$ library (i.e., optimized Computer Vision-Deep Neural Network) (Bradski 2000) for faster and simple detection usage. As shown previously in Figure 11, we localize and classify the melon as compared to negative objects (i.e., background, small melons objects, and plant's oragans) the detected melon image is cropped and prepared to be mapped to the image segmentation step.

To test the accuracy of the predicted bounding box, we used the Intersection over Union (IoU) Equation (5). Using this method, we compare the Ground-truth (GT) boxes (i.e., the boxes that were specified manually) versus the predicted boxes by our model. Figure 13 are examples of applying Equation (5) on multiple images picked randomly from the test dataset, where the green rectangle is the GT box, and the red one is the predicted box. The IoU for all images is more than 0.9, so the accuracy of the detection is excellent regarding Rezatofighi *et al.* (2019).

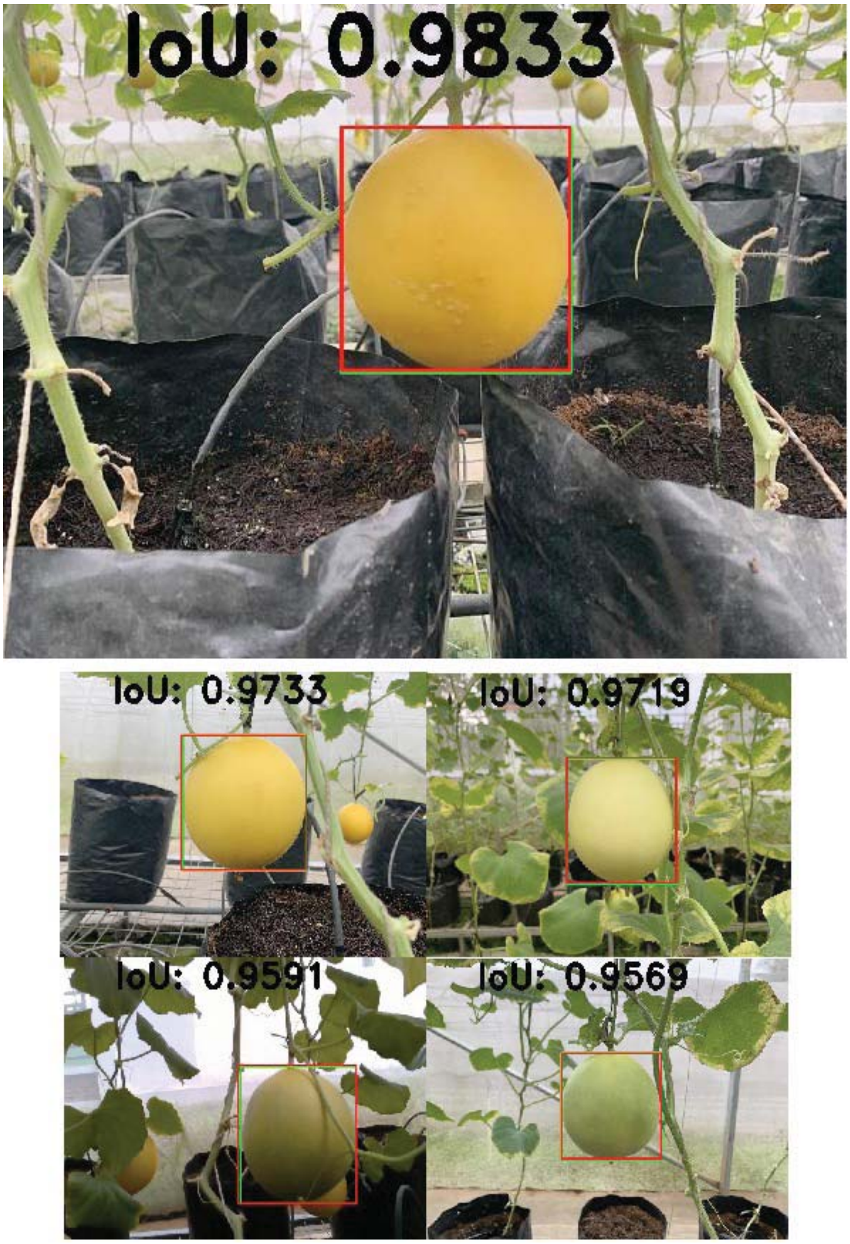


Figure 13 The implementation of the SSD with melon detection and testing the accuracy

4.2 Skin Color Extraction

We have studied, analyzed, and explored the image color using HSV color space to find the color range that has the ROI. Then we were converting the RGB image to HSV. We have created a group of HSV ranges for the available melon categories. Then the detected melon will pass through the following HSV ranges,

1. Green range mask = is the HSV Range of min Green and max Green.
2. Yellow range mask = is the HSV Range of (min Yellow and max Yellow.
3. Orange range mask = is the HSV Range of min Orange and max Orange.

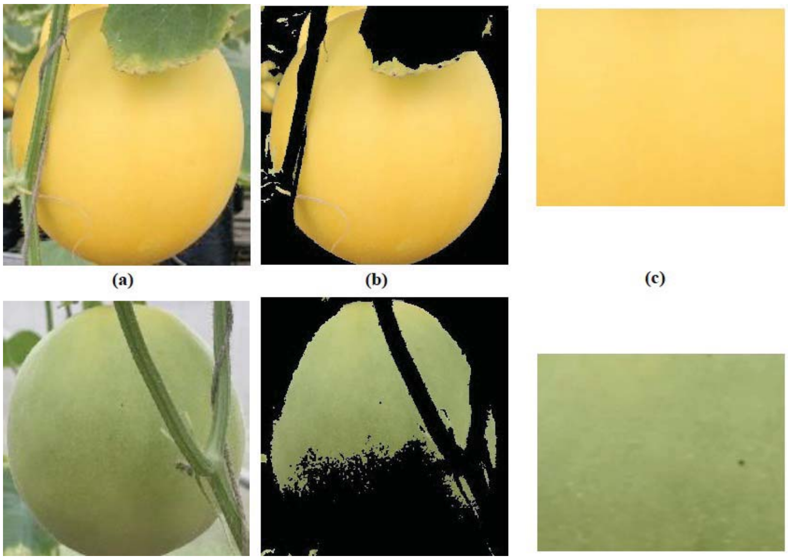
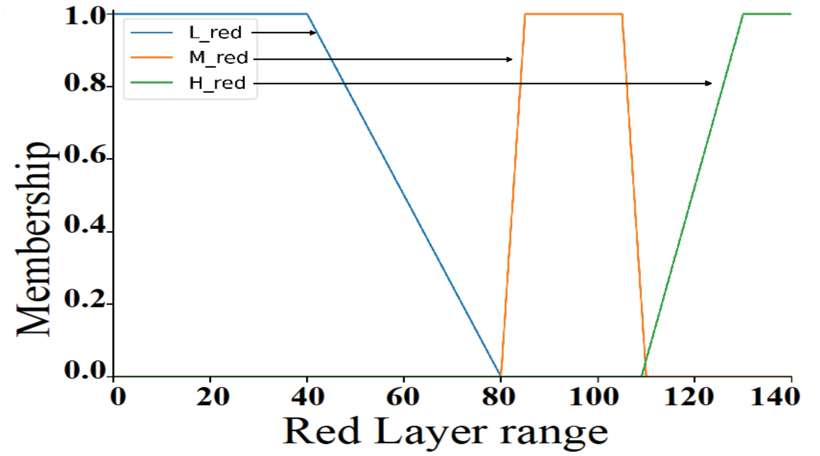


Figure 14 The segmentation-based HSV color space for a ripe orangey melon covered partially with stem and left on the left side, and a green baby melon blocked with plant stem on the right side. The image in level (c) was extracted by taking the max mask for the detected melon

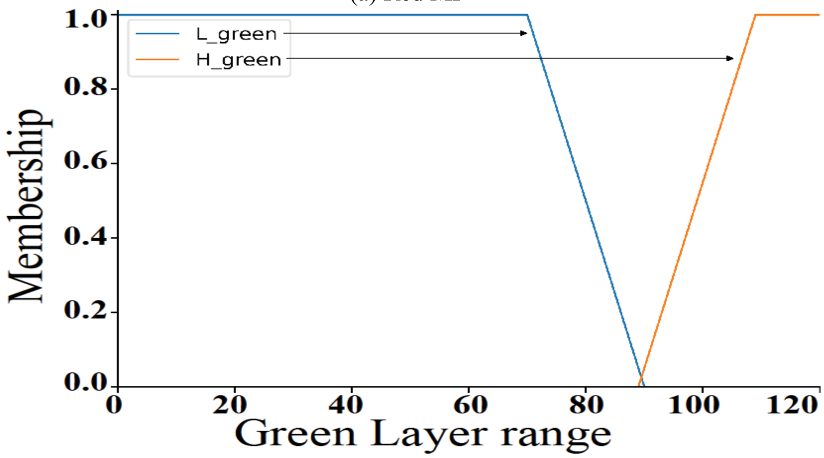
Here we obtained three ranges for the detected melon. For example, if the melon was under-ripe (i.e., green melon's color), then the most extensive area of the image will be the green one, and the same case for yellow or orange melon color Figure 14. The purpose of the masking is to remove the unnecessary background and noise, such as the plant's root, stem, or leaf that overlapping with the melon target. Thus, the remaining area will be considered an ROI candidate Figure 14 (a) and (b). Then it will consider the most extensive available area as shown in Figure 14 (b) and (c). The largest mask area among those three masks will be used as the final mask. An area (square, with size 240×240 as shown in Figure 14 (c) is taken from the final mask, and thus it is used for the next step.

4.3 Fuzzy Logic for Classifying Melon Ripeness

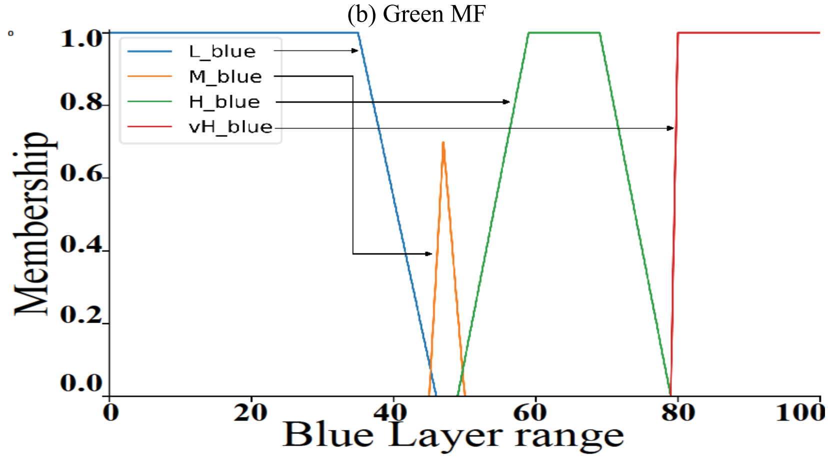
In this section, we will explain and discuss the fuzzy logic system as follow,



(a) Red MF



(b) Green MF



(c) Blue MF

Figure 15 The membership functions of the inputs, (a) are the MF of red, (b) is the MF of green, and (c) is the MF of blue. Note that the (L, M, H, and vH) that MFs names are meaning (Low, Medium, High, and Very High, respectively)

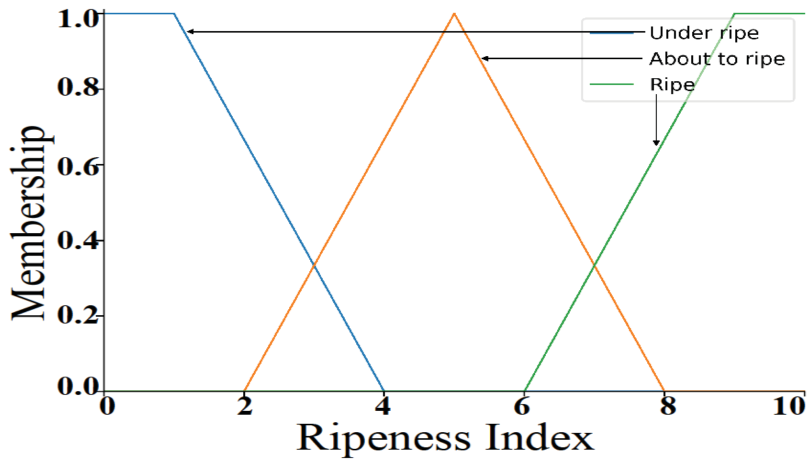


Figure 16 The Defuzzification output membership function. The (blue, orange, and green) lines represent MFs output of Under-ripe, About to Ripe, and Ripe, respectively

The input set of crisp values is the Mean of the normalized *r, g*, and *b* intensities. The reference values for each category are the min-max range of all data, as shown in Table 2 below (red, green, and blue means values). The reference values showed in Table 2 have been extracted from images captured by an iPhone camera. Thus, using other cameras or different light environments may change the RGB pixel intensities. Then, we should modify the reference values regarding the cameras/lightness environment. Otherwise, this will lead to lousy classification.

Table 2 Minimum and Maximum of RGB layers mean values

CATEGORY	RED		GREEN		BLUE	
	Min	**Max**	**Min**	**Max**	**Min**	**Max**
ABOUT TO RIPE	94	102	96	106	50	62
RIPE	126	138	90	103	21	37
UNDER RIPE	92	95	100	104	56	62

In the case was ripe melon, and its blue channel value is more than the Max reference of blue for ripe melon, then it will consider About to Ripe. To get rid of such a problem, we multiply the Medium-blue-MF (M_blue) by 0.7 to decrease the MF output as shown in Figure 15(c) M_blue MF), thus, ignoring the blue when compared to red by using the OR operator Equation (18) below, Jang *et al.* (1996).

$$Ripe\ using\ (OR) = MAX(M\_blue, red) \tag{18}$$

In this case, the red MF will be the winner because the M_blue MF will never reach 0.8. In comparison, red values are always High-red (i.e., Figure 15 (a) H_red MF) in the ripe category. For example, samples of the used rules to classify the melon categories are shown below:

1. If (red is low) and (blue is very high) then (Melon is Under Ripe).
2. If ((red is high) or (green is high)) and (Blue is high) then (Melon is About to Ripe).

3. If ((red is medium) or (green is high)) and (Blue is medium) then (Melon is About to Ripe).
4. If (blue is low) and (red is high) then (Melon is Ripe)
5. If (blue is medium) or (red is high), then (Melon is Ripe). Refer to (18).

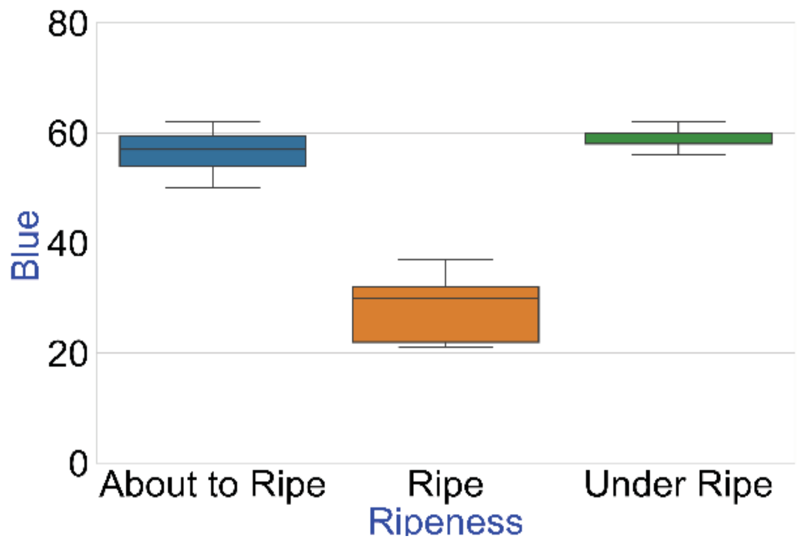
The above rules will determine the melon category and show the results in Figure 16, the Defuzzification output in the range of 0 to 10, calculated using the centroid method. The Defuzzification output explained as follow:

1. Less than 2, the melons are Under Ripe.
2. From 2 to 6, the melons are About to Ripe.
3. More than 6, melons are Ripe.

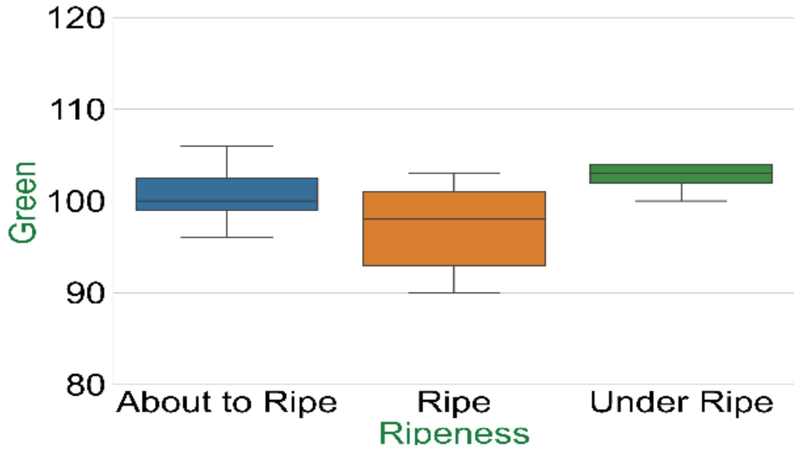
For the sake of testing our system, we captured new 50 images for the same melons categories of the new planting in the same greenhouse on 20/February/2021. We used the same distance from the camera and camera direction mentioned earlier in the Data section, but a different phone camera (i.e., rear iPhone XS camera, 12-megapixel and default image size 4032×3024). Also, we picked 31 labeled images from our previous dataset (i.e., used for training Melon detection). The images used for testing the output of the fuzzy logic system are 81 images in total.

Box-plot created for each channel to find the range of each category. The box-plot explained as follow:

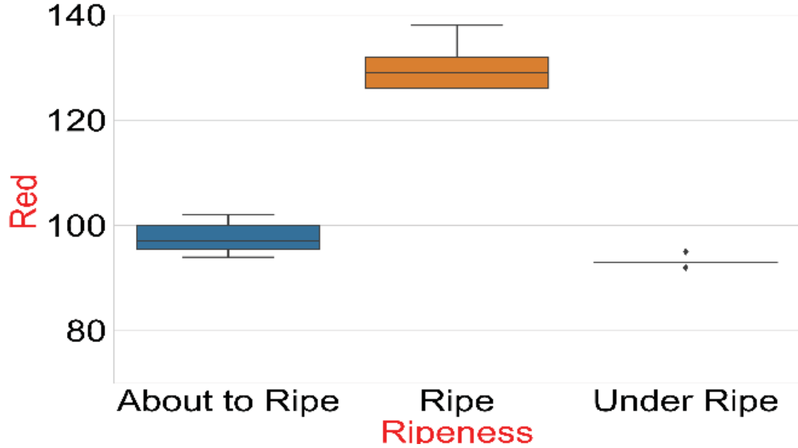
1. Figure 17 (a) Blue is the normalized blue means values for all melon categories.
2. Figure 17 (b) Green is the normalized green means values for all melon categories.
3. Figure 17 (c) Red is the normalized red means values for all melon categories.



(a) Blue



(b) Green



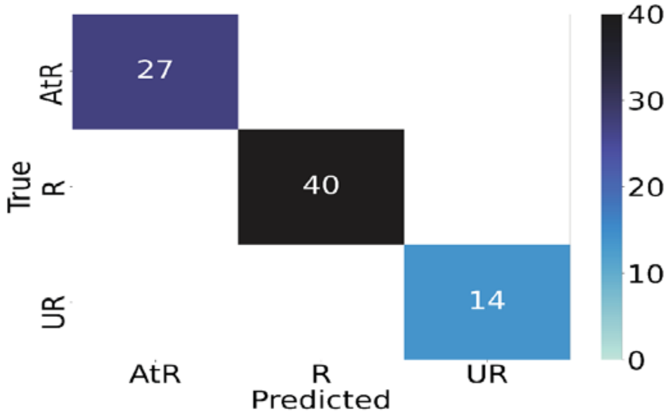
(c) Red

Figure 17 (a) is the blue mean values, (b) is the green mean values, and (c) is the red mean values

We noticed that blue and red highly correlated with the maturity level. You can see in Figure 17 ((a) blue) and ((c) red) the ripe category has the lowest blue values and highest red values, whereas the opposite for the under-ripe category. The magic of detecting the about to ripe category is by far the difference between green and red. The green value is less than red and the opposite in the under-ripe. Also, this category has a slightly lesser red than ripe, whereas blue is high. We can sum up all these in the following points:

- Under Ripe: b is high, and $g > r$, r is low.
- About to Ripe: b is high, $r > g$, r is average.
- Ripe: b is shallow, and r is the highest in all categories.

The testing procedure is simply by creating a loop function to iterate over the extracted melon skin color images labeled with their actual categories and input each image value to the fuzzy logic system. Then, producing a data frame for the actual labels and fuzzy logic output. We used a confusion matrix to compare the system output versus the actual labels to measure the accuracy.



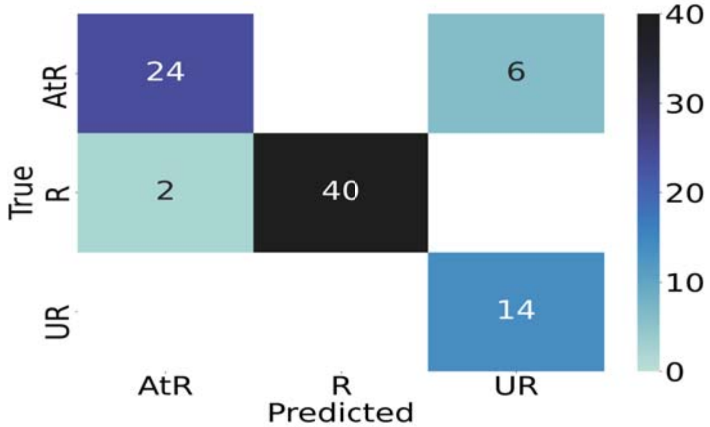
(a) A heat-map plot for image captured by iPhone XS rear camera including all melons categories, and images captured by Intel Realsense D435 camera for ripe melons only

(b) Sample images captured by iPhone XS rear camera

Figure 18 A heat-map plot (a) to describe the accuracy using the same camera used in setting the reference values. The actual melon labels with their corresponding categories on the True axis and the system output on the Predicted axis

As shown in Figure 18 (a), we get a 100% accuracy in classifying (Ripe (R), About to Ripe (AtR), and Under Ripe (UR)) melons. This high accuracy is because the fuzzy rules and the MFs created are based on the reference values in Table 2 (values extracted from images captured by iPhone 6S and iPhone XS rear cameras).

However, AtR has close similarities to R in color, thus, using a different camera leads to less accuracy in classifying AtR versus (UR or R) because of the different camera lens that causes more or less than the range of the references values.



(a) A heat-map plot for the same images used in this study including images captured by Intel Realsense D435 camera for all melons categories

(b) Sample images captured by Intel Realsense D435 camera for the same melon type mentioned in this study

Figure 19 A heat-map plot to show the decreasing in the classification accuracy when using different image source (Intel Realsense D435 camera)

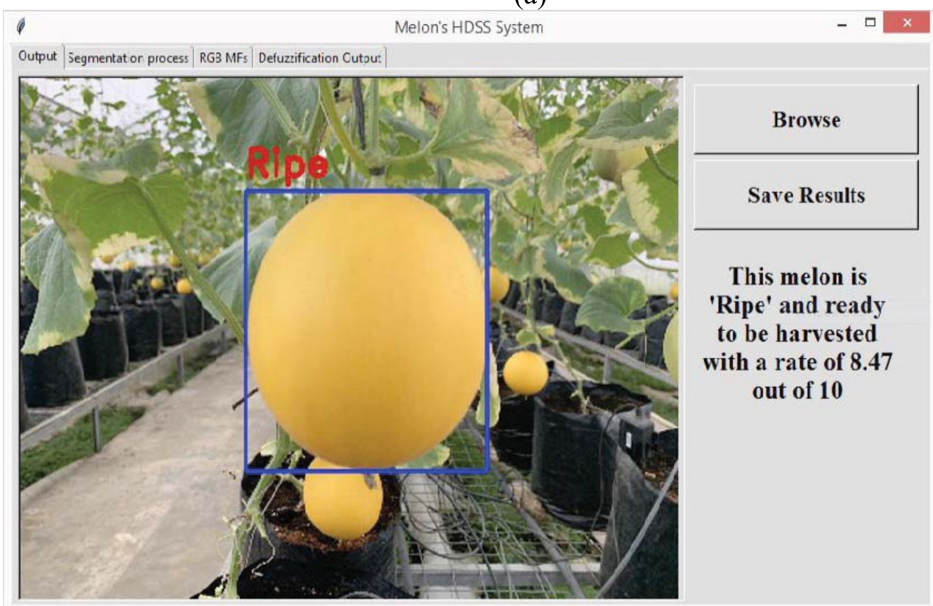
We have tried adding multiple images from various sources such as Intel Realsense D435 camera, the accuracy of classifying UR and AtR becomes 91%, as shown in Figure 19 (a). Still, in general, the ripe category was classified accurately due to the Red layer values were its the highest in this melon category. The misclassification between (AtR and R) can be decreased by using Equation (18) as mention previously, selecting the max MF (M-blue Figure 15(c) OR H_red Figure 15(a)). However, different light environments may or may not leads to less classification accuracy. The accuracy in classifying (UR and R) will still be good because of the vast color differences between the ripe and under-ripe.

As a result, we build a Graphical User Interface (GUI) to help farmers and scientists manage the harvesting of Alisha F1 melon fruits. The GUI is designed using the Tkinter package. It is a Python GUI designer interface (Lundh 1999), and tutorials in CodersLegacy (https://coderslegacy.com).

The usage is by choosing an image for melon fruit by clicking the Browse button. Then the system will display the result in:

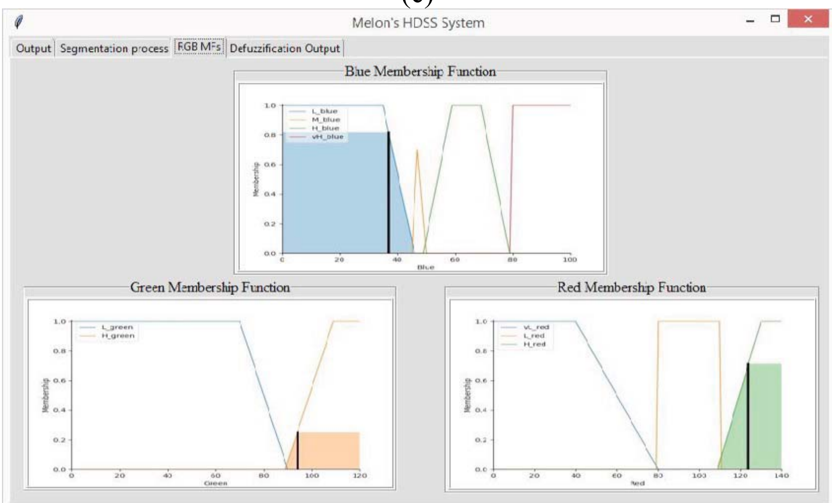
1. The first window displays the detected melon labeled with its category name Figure 20 (a).
2. The second window displays the segmentation process Figure 20 (b).
3. The third window displays the MFs for *r, g* and *b* Figure 20 (c).
4. The last window shows the melon level in the Defuzzification output Figure 20 (d).

(a)



(b)

(c)



(d)

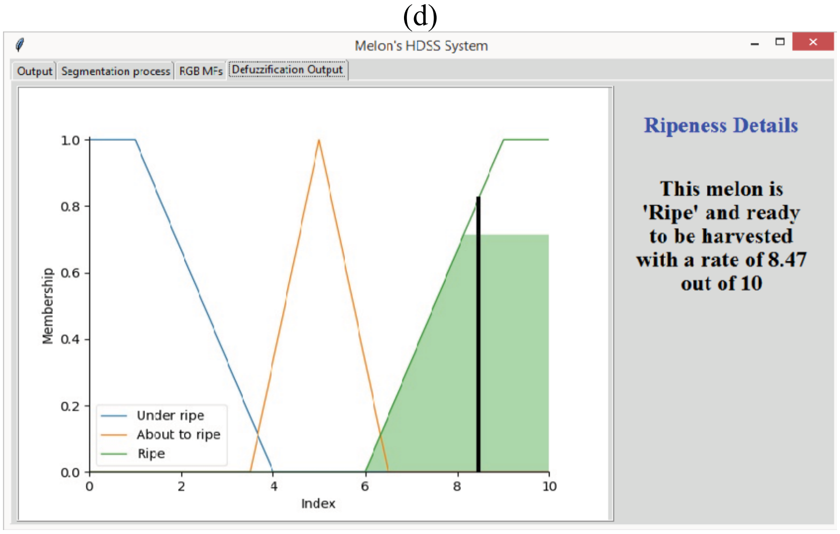


Figure 20 Shows the output on the GUI of the DSS for melon harvesting (M-HDSS)

V CONCLUSION AND SUGGESTION

5.1 Conclusion

A novel detection system of melon's fruit harvesting time has been developed in this study by detecting the melon on the plant, applying image processing technique on the detected melon, extract melon's color, and finally using fuzzy Logic to classify melon fruit into ripe, about to ripe, and under-ripe categories. We achieved 100% classification accuracy by using iPhone 6S rear camera as an image source. By using other image sources, the classification accuracy decreased to 91%. Thus, modifying the reference value for various image sources is an essential factor. The system was developed by using python language, and it can be adjusted to fit specific purpose if is related to a fruit classification problem based on RGB color space.

5.2 Suggestion

As we mentioned earlier, using another image source will affect the classification accuracy due to environmental lightness or different camera specifications. Thus, to make the reference values used for different images source and lightness, we have to find another way to normalize the intensities of the pixels, such as using reference images to apply the normalization based on it. The future work may include an enhancement and improvement to image normalization, as well as detecting melon diseases based on their symptom on the fruit.

ACKNOWLEDGMENT

The data used are images of honeydew melon *Cucumis melo* L. cv. Alisha F1. The images used in this study were captured using a phone camera at Agribusiness and Technology Park (ATP), Cikarawang greenhouse, owned by IPB University in Bogor, Indonesia.

The authors appreciate the Head of the Field Laboratory Sub-Directorate, Directorate of Business Development and Entrepreneurship, IPB University, Dr. Dwi Guntoro, and ATP Manager of IPB University, Mr. Sarwono, to collect the data and the information provided about melon fruit. Without your support, this feat would not be a reality.

REFERENCES

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, *et al.* 2016. TensorFlow: A system for large-scale machine learning. Di dalam: *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*. savannah, GA, USA: USENIX. hlm 265–283. [accessed 2021 Mar 14]. https://tensorflow.org.

Ahmad U. 2017. The use of color distribution analysis for ripeness prediction of Golden Apollo melon.

Amendola M, Neto M, Cruz V. 2005. Using fuzzy sets theory to analyse environmental conditions in order to improve animal productivity. *World*

Congr …. 15 July:29–40. http://www.ime.unicamp.br/~biomat/bio15art3.pdf.

Bange MP, Deutscher SA, Larsen D, Linsley D, Whiteside S. 2004. A handheld decision support system to facilitate improved insect pest management in Australian cotton systems. *Comput Electron Agric*. 43(2):131–147. doi:10.1016/j.compag.2003.12.003.

Bora DJ, Gupta AK. 2014. A New Approach towards Clustering based Color Image Segmentation. *Int J Comput Appl*. 107(12):975–8887.

Bora DJ, Gupta AK, Khan FA. 2015. Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation. 5(2):192–203. http://arxiv.org/abs/1506.01472.

Bradski G. 2000. Open Source Computer Vision Library. *Dr Dobb's J Softw Tools*., siap terbit. https://github.com/opencv/opencv.

Chen C, Huang Y, Li Y, Chen Y. 2021. Identification of Fruit Tree Pests With Deep Learning on Embedded Drone to Achieve Accurate Pesticide Spraying. 9.

Coelho CMM, Bellato C de M, Santos JCP, Ortega EMM, Tsai SM. 2007. Effect of phytate and storage conditions on the development of the ' hard-to-cook .' *J Sci Food Agric*. 1243 December 2007:1237–1243. doi:10.1002/jsfa.

Dadwal M, Banga VK. 2012. Estimate Ripeness Level of fruits Using RGB Color Space and Fuzzy Logic Technique. *Int J Eng Adv Technol*. 2(1):225–229. http://www.ijeat.org/attachments/File/v2i1/A0803102112.pdf.

Freilich S, Lev S, Gonda I, Reuveni E, Portnoy V, Oren E, Lohse M, Galpaz N, Bar E, Tzuri G, *et al.* 2015. Systems approach for exploring the intricate associations between sweetness, color and aroma in melon fruits. *BMC Plant Biol*. 15(1):1–16. doi:10.1186/s12870-015-0449-x.

Galvez RL, Bandala AA, Dadios EP, Vicerra RRP, Maningo JMZ. 2019. Object Detection Using Convolutional Neural Networks. *IEEE Reg 10 Annu Int Conf Proceedings/TENCON*. 2018-Octob May 2019:2023–2027. doi:10.1109/TENCON.2018.8650517.

Ghoury S, Sungur C, Durdu A. 2019. Real-Time Diseases Detection of Grape and Grape Leaves using Faster R-CNN and SSD MobileNet Architectures. Icatces:39–44.

Google. Colaboratory – Google. [accessed 2021 Mar 23]. https://research.google.com/colaboratory/faq.html.

Hartati S, Sitanggang IS. 2010. A fuzzy based decision support system for evaluating land suitability and selecting crops. *J Comput Sci*. 6(4):417–424. doi:10.3844/jcssp.2010.417.424.

HSV color solid cylinder saturation gray.png - Wikimedia Commons. [accessed 2021 Mar 26]. https://commons.wikimedia.org/wiki/File:HSV_color_solid_cylinder_saturation_gray.png.

Iskandar BA, Suwarno WB, Gunawan E, Saptomo SK. 2019. Selection of potential genotypes and traits evaluation of honeydew (Cucumis melo L.). *IOP Conf Ser Earth Environ Sci*. 383(1). doi:10.1088/1755-1315/383/1/012012.

Jang J-SR, Sun C-T, Mizutani E. 1996. *Neuro-Fuzzy And Soft Computing.pdf*. 1st editio. Prentice Hall India; 1st edition (1996) (1600).

Kays SJ. 1999. Preharvest factors affecting appearance. *Postharvest Biol Technol*. 15(3):233–247. doi:10.1016/S0925-5214(98)00088-X.

Kopáčková H, Škrobáčková M. 2006. Decision support systems or business intelligence : what can help in decision making? January 2006.

Kumar T, Verma K. 2010. A Theory Based on Conversion of RGB image to Gray image. *Int J Comput Appl*. 7(2):5–12. doi:10.5120/1140-1493.

Li M, Sui R, Meng Y, Yan H. 2019. A real-time fuzzy decision support system for alfalfa irrigation. *Comput Electron Agric*. 163 April:104870. doi:10.1016/j.compag.2019.104870.

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. 2014. Microsoft COCO: Common objects in context. Di dalam: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Volume 8693 LNCS. Springer Verlag. hlm 740–755.

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. 2016. SSD: Single shot multibox detector. Di dalam: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Volume 9905 LNCS. Springer Verlag. hlm 21–37.

Lundh F. 1999. An Introduction to Tkinter. http://www.pythonware.com/library/tkinter/introduction/index.htm.

Marques O. 2011a. 299-335morph Practical Image and Video Processing Using MATLAB - Google Books. John Wiley & Sons. hlm 299–335. [accessed 2021 Mar 22]. https://books.google.co.id/books?hl=en&lr=&id=xzD25QEo8qYC&oi=fnd&pg=PR21&dq=gonzalez+image+processing+using+matlab&ots=45Lf9ETct7&sig=eOy2p-5Tci2DA5f0Ci2QLDnUjwM&redir_esc=y#v=onepage&q=Erosion&f=false.

Marques O. 2011b. Practical Image and Video Processing Using MATLAB - Google Books. Di dalam: *Practical Image and Video Processing Using MATLAB*. John Wiley & Sons. hlm 387–425. [accessed 2021 Mar 21]. https://books.google.co.id/books?hl=en&lr=&id=xzD25QEo8qYC&oi=fnd&pg=PR21&dq=gonzalez+image+processing+using+matlab&ots=45Lf9ETct7&sig=eOy2p-5Tci2DA5f0Ci2QLDnUjwM&redir_esc=y#v=onepage&q&f=false.

Pedrycz W, Sillitti A, Succi G. 2016. Computational intelligence: An introduction. *Stud Comput Intell*. 617:13–31. doi:10.1007/978-3-319-25964-2_2.

Piyush S, Dhrumil S, Bhavikkumar P. 2014. Temperature Control using Fuzzy Logic. *Int J Instrum Control Syst*. 4(1):1–10. doi:10.5121/ijics.2014.4101.

Power DJ. 2002. *Decision Support Systems: Concepts and Resources for Managers - Daniel J. Power - Google Books*. illustrated. Greenwood Publishing Group, 2002. [accessed 2020 Sep 11]. https://books.google.co.id/books?id=9NA6QMcte3cC&source=gbs_navlinks_s.

Python GUI with Tkinter - CodersLegacy. [accessed 2021 Mar 21]. https://coderslegacy.com/python/python-gui/.

Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. Di dalam: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Volume ke-2019-June. hlm 658–666.

Simonyan K, Zisserman A. 2015. Very deep convolutional networks for large-scale image recognition. Di dalam: *3rd International Conference on Learning*

Representations, ICLR 2015 - Conference Track Proceedings. International Conference on Learning Representations, ICLR. [accessed 2021 Mar 28]. http://www.robots.ox.ac.uk/.

Sural S, Qian G, Pramanik S. 2002. Segmentation and histogram generation using the HSV color space for image retrieval. *IEEE Int Conf Image Process*. 2:589–592. doi:10.1109/icip.2002.1040019.

Tzutalin. 2015. labelImg. [accessed 2021 Mar 19]. https://github.com/tzutalin/labelImg.

Venkatalakshmi B, Devi P. 2014. Decision Support System for Precision Agriculture. *Int J Res Eng Technol*. 03(19):849–852. doi:10.15623/ijret.2014.0319154.

Wang H, Qiu D. 2003. Computing with Words via Turing Machines: A Formal Approach. *IEEE Trans Fuzzy Syst*. 11(6):742–753. doi:10.1109/TFUZZ.2003.819841.

WBG. Agriculture | International Development Association - World Bank | International Development Association - World Bank. [accessed 2021 Apr 12]. http://ida.worldbank.org/results/agriculture.

Zadeh LA. 1997. Roles of soft computing and fuzzy logic in the conception, design and deployment of intelligent systems. *IEEE Int Conf Fuzzy Syst*. 1(4):1. doi:10.1007/978-3-642-58930-0_1.

Zhao M, Bu J, Chen C. 2002. <title>Robust background subtraction in HSV color space</title>. *Multimed Syst Appl V*. 4861:325–332. doi:10.1117/12.456333.

Zheng S, Guan W, Li B. 2015. Digital Display Design of Ethnic Clothing of Nanling. Icmmcce:2805–2808. doi:10.2991/icmmcce-15.2015.541.

Zheng S, Guan W, Li B, QIN D. 2016. Analysis of Internet of Things Talent Training and Curriculum System Innovation. Icemct:957–960. doi:10.2991/icemct-16.2016.208.

AUTHOR BIOGRAPHIES

Jaafar Mohammed Jabbar is currently a Master's degree student at the Department of Computer Science, IPB University, Bogor, Indonesia. His bachelor's degree in Computer Science, AL-Rafidain University College, Baghdad, Iraq. He worked as a programmer at the Iraqi Industrial Ministry, Baghdad, Iraq. His interests, computer vision, database management, image processing, and machine learning. Jaafar became a member of the IAENG in 2021.