DIP-CBML: A New Classification of Thai Dragon Fruit Species from Images

Naruwan Yusamran, Nualsawat Hiransakolwong
Department of Computer Science-School of Science, King Mongkut's Institute of Technology, Ladkrabang
Bangkok, Thailand.

***Abstract*—The attractiveness of dragon fruit is that it has a strange exterior, beautiful colors, and high nutritional value. In Thailand, there is both import and export of dragon fruit. Each package for export must contain only one species of dragon fruit. From the survey, there are seven species of dragon fruit cultivated in Thailand and only some farmers can identify them on his/her farm. Therefore, this research focuses on the classification of Thai dragon fruit from laboratory images and outdoor images; which is different from the previous works which studied only laboratory images. This method was named DIP-CBML that stands for digital image processing with content-based and machine learning. The method consists of image type identification, pre-processing, red and yellow classification, image background removal, and six classes of red species classification. From the results, DIP-CBML can work with both datasets. It gave 100%, 100% and 95.53% accuracy for the image type identification, red and yellow classification, and the classification of six red species respectively. Hopefully, this research will lead to the innovation for the pre-harvest classification of Thai dragon fruit cultivars, applied to industrial applications, and robot harvesting. In the future, may add value to the yield of Thai dragon fruit cultivation.**

***Keywords*—Dragon fruit; classification model; outdoor dataset; image pre-processing; segmentation**

I. Introduction

Dragon Fruit is a fruit in the cactus family (Cactaceae) with genus Hylocercus Spp. and Selenicereus Sp. It has been planted in Thailand since 1997. With its unusual exterior and beautiful colors, it has received much attention. The advantage of dragon fruit is that it is a fruit that has high nutritional value. It can be used in the food, pharmaceutical, and cosmetic industries [1-4]. Since dragon fruit is a juicy fruit, it will be bruising and perishable in a short time after harvesting. It must be eaten or processed quickly, or storing at low temperatures for slowing down spoilage [5].

Currently, there is import and export of dragon fruit. Therefore, the agricultural standard for dragon fruit has been established by the National Bureau of Agricultural Commodity and Food Standards (ACFS), the Ministry of Agriculture, and Cooperatives of Thailand. Dragon fruit is divided into three groups by the peel color and the pulp color, namely Group 1, red peel with white pulp (Hylocereus undatus), Group 2, red peel with red pulp (Hylocereus polyrhizus) or pink pulp (Hylocereus spp.) and Group 3, yellow or gold peel with white pulp (Hylocereus sp. and Selenicereus sp.). The package must contain the same dragon fruit species per package [6].

From a survey on the cultivation of dragon fruit in Thailand. Loei Province located at the top of the Northeast of Thailand is the area that can harvest the most products in the country [7]. It was found that there are seven species of Thai dragon fruits planted. If divided like ACFS, the First group has two species, which are Jumbo White and Vietnamese White. The second group has four species, which are Pink, Siam Red, Taiwan Red, and Ruby Red. The third group has only one species called Israel Yellow. Each species has different care and maintenance which affects cultivation costs and selling prices.

The main problem is that only some farmers can identify the cultivars which are grown. Like the middleman, most farmers only know the color of the peel and of the pulp. As a result, the price of each species of dragon fruit is determined according to the standard in only three groups. If farmers or middlemen can classify dragon fruit species, they know how to manage production of dragon fruit, take care the produce before and after harvesting, and manage the transportation according to the specific morphology of each species. This will affect the cost and selling price, and also promote the conservation of fruit species.

However, species identification based on morphology requires a great deal of knowledge and expertise. For the general public new farmers and middlemen can identify dragon fruit species accurately and easily, the researcher has researched only the laboratory dataset [8,9] which gave a high accuracy of 98%.

Therefore, this research will present the classification of Thai dragon fruit species from images in both indoor (laboratory) and outdoor datasets. The research conducted experiments with image processing techniques, content-based techniques, machine learning, and deep learning. In the future, hopefully, this will lead to the development of innovations for the classification of Thai dragon fruit species pre or post harvesting that can be easy to use, convenient, and suitable for people. This can be applied to industrial applications and robot harvesting. It may add value to the yield of Thai dragon fruit cultivation. This paper has presented Theories and Literature Reviews in Section II, The DIP-CBML Method in Section III, Experiment Results in Section IV, Discussion in Section V, Conclusions in Section VI and Suggestion in Section VII.

II. Theories and Literature Reviews

A. Color Model

Color Model is a color system that controls the display of digital images on a digital device. Each type of color model has its own way of generating colors that have different structures. Therefore, different types of color models are used for different purposes [10]. The most popular color models in image processing are the RGB, HSV, and Lab model.

The RGB is a color system formed by the combination of primary colors. There are Red, Green, and Blue which are channels of color digital images. Each channel has 0-255 color levels [11].

The HSV is a color model which is close to the color perception of human vision. It consists of Hue (H), Saturation (S), and Value (V). Hue is the color tone defined by the angle of the color wheel. In image processing by OpenCV, the H value ranged from 0° to 180°. The S is the saturation of the color tone. If S is 0, the white color is always displayed. The V is the brightness of the tone. If the brightness value is 0, the result will be black. In OpenCV processing, S and V values ranged from 0 to 255. If both values are set to 255, it will display a pure color tone [12, 13].

The Lab color model consists of three channels. There are the L channel, a channel, and b channel. L channel is the lightness value, an a channel is a green gradation to red, b channel is a blue gradation to yellow. In OpenCV, Lab values are scaled in the range from 0 to 255 for all three channels [11, 13].

B. Digital Image Processing

Digital image processing is a process that uses a digital image as input for a specified processing purpose, such as image resizing, converting a color image to grayscale, data augmentation, etc. A digital image can be represented by F(x,y) where x and y are spatial coordinates. Each coordinate is called a pixel. There are three types of images: color images, grayscale images, and binary images [14].

C. Image Pre-processes

Image pre-processing is an important step in digital image processing because the raw data from the acquisition, which is interpolated with multifactorial noise, cannot be processed immediately. Therefore, images must be done with image pre-processes to reduce those noise. For example, image segmentation is a technique that helps extract and segment desired or unwanted image data. It helps to get more specific information and it is easier to analyze information. However, processed data must be ensured that the remaining information is correct. There is no excess or missing [15]. Techniques used in image segmentation are the segmentation by thresholding-based, edge-based, area-based, and energy-based [16-18]. There are also other techniques to prepare data such as intensity estimation, geometric estimation, elementary processing, holistic processing, etc. [15].

D. Feature Extraction

Feature extraction is the process of extracting important characteristics of an image to describe all information in the image. The feature descriptor may be all or part of the image data that can represent the whole image. In general, attributes about color, shape, size, and texture are used in image analysis or image classification. Of course, image data may not have only one attribute that can identify specific characteristics. For Image classification by machine learning, the image features will be represented by a feature vector that is a 1-dimensional array [14, 15]. In contrast for deep learning, the feature extraction process is performed automatically. The researchers are responsible for specifying the desired feature map size, making it very convenient for researchers.

E. Image Classification Model

The Image Classification Model is a classifier that was created to identify or classify groups of image data. The classifiers use rule-based classifiers which are processed under a "condition" that is defined in the IF condition THEN conclusion form. Image classification model with a content-based feature in which researchers must extract features for suit methods such as support vector machine (SVM), k-nearest neighbor (KNN) and decision trees (DT), etc. [19-24].

The deep learning model is the popular automated method. The model gives high predictive performance [25] by mimicking the operation of human neurons by building a neural network with multiple nonlinear processing layers [14, 26]. Each layer takes the result of the previous layer as input. The strength of deep learning is that it supports the extraction of ambiguous features. The model extracts feature automatically. The researcher is responsible for preparing the data that will be used for learning only. It is very convenient for researchers.

F. Literature Reviews

This research focuses on reviewing literature related to fruit classification to study methods that can be used to classify Thai dragon fruit from images. Literature reviews are found that most researchers carried out research in four major steps: data acquisition, data pre-processes, feature extraction, and classification [15, 27]. Image datasets used include both image datasets created in the laboratory [22, 24, 28-31] and image datasets created at the outdoor environment [20, 23, 32-37]. Of course, image datasets from outdoor environments are more complex than those created in laboratories. Therefore, the steps to process this type are also more complicated.

Most researchers focused on image pre-processing like image segmentation techniques to prepare image data differently. Either the graph cut method [20], the conversion of a color image to a binary image by calculating the threshold value [8, 29], or using machine learning to segment the fruit from the background. This can be applied to the detection of fruit on the tree [23, 33, 34, 36, 37]. It reduces the complexity of the image and improves efficiency in feature extraction and classification. Muresan and Oltean [28] has proposed the use of threshold values in image segmentation with the HSV color model to separate the dragon fruit from the background of the plant. The objective is to detect and count the amount of dragon fruit on the tree by the image dataset of Hylocereus Undatus dragon fruit species in the outdoor environment. The accuracy is more than 80%.

Jana. et al. [20] has presented the recognition of fruits from the natural image in eight types. The image data was prepared by the graph cut method, which can separate the background from the fruit very well. After the image data preparation step, color and texture features were used for fruit recognition with the support vector machine (SVM) algorithm, giving an accuracy of 83.33%.

Muhammad [22] presented the use of color, texture, shape, and size attributes in classifying four date palm species by histogram of the local binary pattern (LBP) and Weber local descriptor (WLD) data. The Fisher discrimination ratio (FDR) was used to select the top 10 most important features that gave the greatest FDR. For shape and size features, the image data were analyzed from four features, namely Major axis length, Minor axis length, Ellipse eccentricity, and Area. In total, there are 14 features with a support vector machine (SVM) which gives an accuracy of up to 99%.

Fu. et al. [23] has presented the detection of bananas on the tree with machine learning. There is an image data pre-processing to reduce the complexity of the image by using Otsu's algorithm to find threshold values for converting HSV images to binary images. In the detection process, a histogram of oriented gradients (HOG) and local binary patterns (LBP) were used to extract the shape and texture features of the banana. The experiments found that the algorithm which combines HOG, LBP, and SVM gives maximum efficiency. It has an accuracy of 100% for the training set and 89.63% for the test set.

The DIPDEEP method [8] was the classification of Thai dragon fruit from a laboratory images dataset. DIPDEEP used image processing techniques to prepare images based on a threshold value to separate the background from the dragon fruit. Then, the images were used to classify the seven cultivars using rule-based classification techniques to classify dragon fruit groups by skin color and deep learning model based on a 13-layers of convolutional neural network that starts with an RGB image size of 100x100 pixels as an input layer and ends with six output nodes. Optimizer based on Adam algorithm and set Learning rate equal to 0.001, batch size and epoch equal to 100. The accuracy of red and yellow classification was 100%, the accuracy of DIPDEEP was 98.80%. For outdoor images, the results were not satisfactory.

The CBML model [9] was the classification of Thai red dragon fruit from a laboratory images dataset. CBML method used a content-based model that uses a total of 34 attributes, consisting of two attributes of texture features from dissimilarity (D) and asm properties only directions: 0° of GLCM algorithm with RGB image and 32 attributes of color features with RGB, HSV, and AB of Lab color model, for each channel in any color model using four statistical features: mean, standard deviation, skewness, and kurtosis. Each attribute used minmaxscaler to normalize into the 0 to 1 range and uses machine learning for giving optimization results with the support vector machine (SVM), setting kernel for polynomial in 8 degrees. It has an accuracy of 100% for the training set and 98.47% for the test set. The outdoor image was not processed.

Therefore, this paper focused on modeling the classification of Thai dragon fruit species from images that support two types of image datasets; laboratory and outdoor. The required model will support pre- and post-harvest processing. Hopefully, this research will be applied to industrial applications and agriculture technology.

III. THE DIP-CBML METHOD

This paper presents algorithms to classify species of Thai dragon fruits automatically that support two types of image datasets called DIP-CBML which stands for digital image processing with content-based and machine learning. The DIP-CBML will be able to process both types of images when DIP-CBML is able to classify the image types. It makes subsequent processing work correctly according to the characteristics of the two image types which are different processes. If the DIP-CBML can classify the group of dragon fruit from the peel color which is red or yellow, the processing time was reduced because the yellow group takes shorter time to process than the red group. Therefore, the DIP-CBML structure consists of image type identification, pre-processing, red and yellow classification, image background removal, and six classes of red classification as Fig. 1.

A. Datasets

The image datasets consisted of 9,754 laboratory images and 9,067 outdoor images from [8], each image containing one dragon fruit of one species. Both datasets were taken with a mobile phone camera. The 1:1 aspect ratio was set for laboratory recording. Unlike the outdoor images, the aspect ratio is set independently, i.e., 3:4 and 9:16, resulting in different image sizes with any lighting conditions.

In addition, the characteristics of the images are different, because the images in the laboratory are an image of the post-harvest dragon fruit placed in a photographic box and used black velvet fabric in the background as Fig. 2(a), but the outdoor images are the dragon fruit growing on the tree until it is ready to be harvested as Fig. 2(b). Obviously, the outdoor images have more complex backgrounds such as sky, ground, grass, and branches, while the laboratory images have a black background.

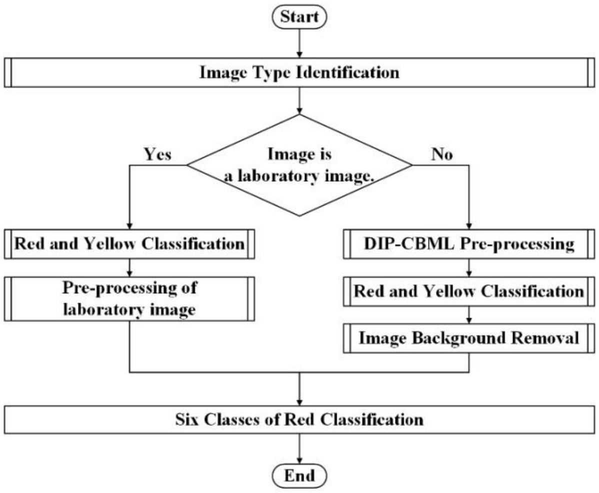
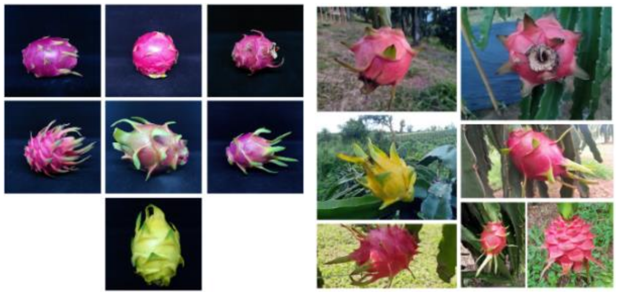

Start
Image Type Identification
Yes
Image is a laboratory image.
No
Red and Yellow Classification
DIP-CBML Pre-processing
Pre-processing of laboratory image
Red and Yellow Classification
Image Background Removal
Six Classes of Red Classification
End

Fig. 1. The Steps of DIP-CBML method.



(a) (b)

Fig. 2. Examples of two datasets as (a) laboratory images and (b) outdoor images.

B. Image Type Identification

This step is used to identify the type of images from either laboratory image or outdoor image because the image data pre-processing of the two types are different. The procedure is shown in algorithm 1.

Algorithm 1: Image Type Identification

```
Input: Original image with RGB color model.
Output: Type of image; laboratory or outdoor.
Resized Image = Resize the input image to not over 300 on each side.
R = R channel of Resized Image.
Gray = Gray scale of Resized Image.
Mean Threshold = (Sum of R value)/(image size)
MKR = The conversion result of R image to binary by Mean
      Threshold.
BG Pixel = The gray pixel at the same coordinate as a black pixel in
           the MKR.
If (BG Pixel != 0)  then
    MP = Mean value of BG Pixel
    STD = Standard Deviation value of BG Pixel
    KS = Kurtosis value of BG pixel
    If (MP < 31.77) then
        Output is a laboratory image type.
    else if (MP > 48.62) then
        Output is an outdoor image type.
    else
        If (STD < 15.27) then
            Output is a laboratory image type.
        else if (STD > 20.52) then
            Output is an outdoor image type.
        else
            If (KS>=8.32 && KS<=54.37) then
                Output is a laboratory image type.
            else
                Output is an outdoor image type.
            End
        End
    End
else
    Output is an outdoor image type.
End
```

From the characteristics of the laboratory images with a flat background, it is not as complex as the outdoor images. Therefore, the three statistical properties of the background pixels are used to identify the type of images. There are the mean, standard deviation, and kurtosis as shown in Fig. 3.

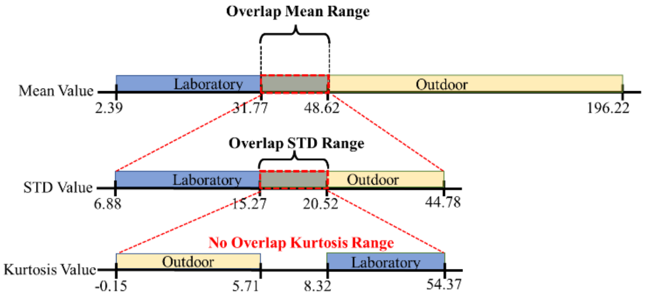


Fig. 3. The statistical properties of each type of image.

From the random 700 samples of each type of image, it was found that the laboratory images had an average of background pixels in a gray level ranging from 2.39-48.62 and outdoor images in ranging 31.77-196.22, because there is an overlap range. When considering the other statistics of the overlapping range, it was found that other statistics such as mode, median, standard deviation and skewness still overlap. Therefore, the statisics value that has the least overlap is the standard deviation. From the overlapping average range, it was found that the laboratory images had a standard deviation of background pixels in a gray level ranging from 6.88-20.52 and outdoor images from 15.27-44.78. Successively, at the overlapping standard deviation range, it was found that a kurtosis of background pixels in a gray level range is not overlapping. The laboratory images had a kurtosis of background pixels in a gray level ranging from -0.15-5.71 and outdoor images from 8.32-54.37.

Therefore, this step begins with the original image being reduced to a size of no more than 300x300 in proportion. Set the R image to represent the R channel of the RGB image and Gray to represent the grayscale image. The R image was used to create a segmentation mask, which is a binary image, using the mean threshold as the threshold value for the binary conversion as equation (1).

$$Mean\ Threshold = (Sum\ of\ R\ value)/(image\ size) \quad (1)$$

If any pixel of the R image is greater than the mean threshold, this pixel will be set to white as the object, otherwise, this pixel will be set to a black background. The morphology function opening is used to decrease noises of the object, the Fill Hole from OpenCV is used to fill in any hole in the object, and the morphology function erosion is used to slightly decrease the edges of the image, respectively. The result is the MKR mask, a binary image with black pixels as the background (BG) as shown in Fig. 4 on the 2rd row.

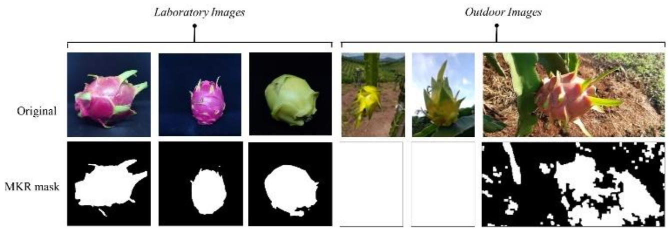


Fig. 4. Characteristics of Laboratory and Outdoor images after coverting to binary mask by using the mean threshold.

For an image where background pixels are not found, it can be concluded that it is an outdoor image. On the other hand, calculate the mean, standard deviation and kurtosis of BG pixels which is a gray pixel at the same coordinate as a black pixel in the MKR and replace mean with MP, replace standard deviation with STD and replace kurtosis with KS.

As mentioned above, the mean of the background pixels of both images overlaps. Therefore, the type of image can be clearly identified. An image with an average less than 31.77 is a laboratory image. An image with an average greater than 48.62 is an outdoor image. If any image is within the overlapping average range of the background pixels which is from 31.77-48.62, consider a standard deviation of the background pixels instead. If a standard deviation of the background pixels less than 15.27, an image is a laboratory image. If a standard deviation of the background pixels greater than 20.52, an image is an outdoor image. As the same, any image is within the overlapping standard deviation range of the background pixels which is from 15.27-20.52, consider a kurtosis of the background pixels instead. If a kurtosis of the background pixels between 8.32 and 54.37, an image is a laboratory image. On the other hand, that image was an outdoor image.

C. DIP-CBML Pre-processing

As mentioned above, the image data pre-processing of the two types are different. The laboratory image can use the process of creating MKR mask as image pre-processing but not for an outdoor image. Therefore, the pre-processing of DIP-CBML was used to support an outdoor image especially. This process still aims to separate the background from the dragon fruit by creating a mask that can be used to divide into two parts. The steps as shown in algorithm 2 and Fig. 5, DIP-CBML uses the graph cut algorithm to create the segmentation mask. It is generally known that graph cut requires a position of interest object covered in a rectangle. Therefore, it needs to locate the dragon fruit before the graph cut algorithm was used to create the segmentation mask.

To locate the dragon fruit, the HSV color model was used to create a binary image that helps locate the fruit in the image by HSV ranging same as [8] which consists of the H range, S range, and V range in the OpenCV system. There are H ranging from 0-23 and 151-179 for a red and 23-37 for a yellow. For S and V ranging from 100-255. The result is a binary image that is created by any pixel in the HSV image within this range will be scaled as a white pixel represent for a fruit pixel. On the contrary, a black pixel is an out of range; represent for a background pixel. Then, bring the binary image to the detect location of dragon fruit subprogram as shown in Fig. 5.

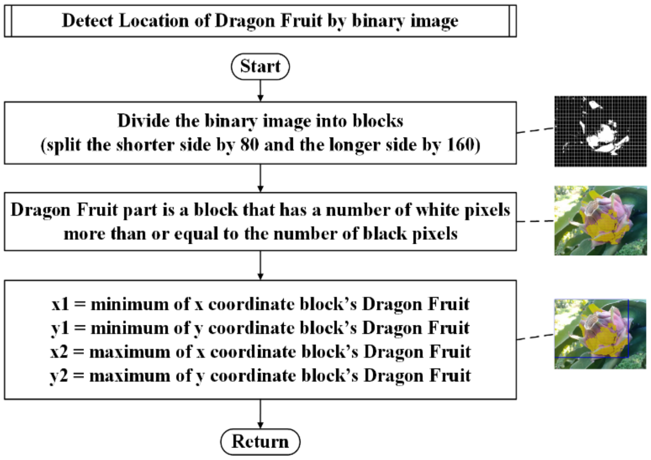

Detect Location of Dragon Fruit by binary image
Start
Divide the binary image into blocks
(split the shorter side by 80 and the longer side by 160)
Dragon Fruit part is a block that has a number of white pixels more than or equal to the number of black pixels
x1 = minimum of x coordinate block's Dragon Fruit
y1 = minimum of y coordinate block's Dragon Fruit
x2 = maximum of x coordinate block's Dragon Fruit
y2 = maximum of y coordinate block's Dragon Fruit
Return

Fig. 5. Flowchart of detecting dragon fruit location.

In Fig. 5, the binary image is divided into grid blocks by that have no more than 160x80 blocks. The size of each block was calculated by dividing the shorter side by 80 and the longer side by 160. For example, the image size is 4,608x2,240 pixels, the longer side was divided, as 4,608/160 = 28.8 approximated 29 , and the shorter side was divided, as 2,240/ 80 = 28. Therefore, each block size is equal to 29x28 pixels. Then, the binary image is segmented into many blocks with 29x28 pixels as the grid block size. From this block size, the number of blocks is equal to 158x80 blocks. The number of blocks is not over 160x80 blocks. Any block with the number of white pixels greater than or equal to the number of black pixels is defined as part of the dragon fruit. The position of all the blocks that are part of the dragon fruit is the location of the fruit. If all blocks are merged into one rectangle then let x1 equal the minimum value of the x-coordinate, y1 equal the minimum value of the y-coordinate, x2 equal the maximum value of the x-coordinate, and y2 equal the maximum value of the y-coordinate. The coordinate (x1,y1) is the northwest corner of the rectangle, and the coordinate (x2,y2) is the southeast corner of the rectangle. The coordinates (x1,y1,x2,y2) will be the boundary used to segment the image with the graph cut, which must not be equal to the border of the image. If the coordinates (x1,y1,x2,y2) is equal to the border of the image, the binary image was modified by the morphology function erosion with kernel size 5x5 and returned to the detect location of dragon fruit processes until the coordinates are not equal to the border of the image. Then, the coordinates (x1,y1,x2,y2) were set as the rectangle that will be used to create the mask by the graph cut algorithm that can separate the background from the dragon fruit.

To create the mask, DIP-CBML uses a graph cut algorithm with the coordinates which were modified to four directions, namely (x1,y1,x2,y2), (0,y1,x2,y2), (x1,0,x2,y2), and (0,0,x2,y2) because the background around the fruit was removed as shown in Fig. 6. The four results are RGB color images, converted them to a mask that is a binary image by setting the part of the dragon fruit to be a white pixel and the background part to be black. Then we bring all four masks to the white pixels together with an intersection mask represented by the HSV1-Mask.

The HSV1-Mask can be used to separate the background from the dragon fruit where black pixels refer to the background and white pixels refer to dragon fruit. Any coordinate in RGB image that has the same coordinates as a background pixel are changed to 0. The result is RGB image with some or all of its background black, represented by the Image-HSV1. The illustration of DIP-CBML pre-processing as shown in Fig. 6 and the result of this step is shown in Fig. 7(b).

Algorithm 2: DIP-CBML pre-processing

```
Input: Original image with RGB color model.
Output: Image-HSV1, HSV1-Mask, and Coordinates (x1,y1,x2,y2).
hsv_img = Result of converting input image RGB to HSV color
          model.
binary_img = Result of converting hsv_image to binary by HSV
             ranges from [8]
(x1,y1,x2,y2) = Return values of Detecting Dragon Fruit Location by
                binary_img
While ((x1,y1,x2,y2) == Border of image) do
    Erosion binary_img with kernel 5x5
    (x1,y1,x2,y2) = Return values of Detecting Dragon Fruit Location
                    by  erosioned binary_img
    End
IMG1, IMG2, IMG3 and IMG4 = Return values of Grab Cut
4Directions by Coordinates (x1,y1,x2,y2).
M1, M2, M3 and M4 = Return values of Converting IMG1, IMG2,
                    IMG3 and IMG4 to mask
HSV1-Mask = Result of Intersection all Masks (M1, M2, M3 and M4)
Image-HSV1 = Output of Removing Background by HSV1 Mask
```

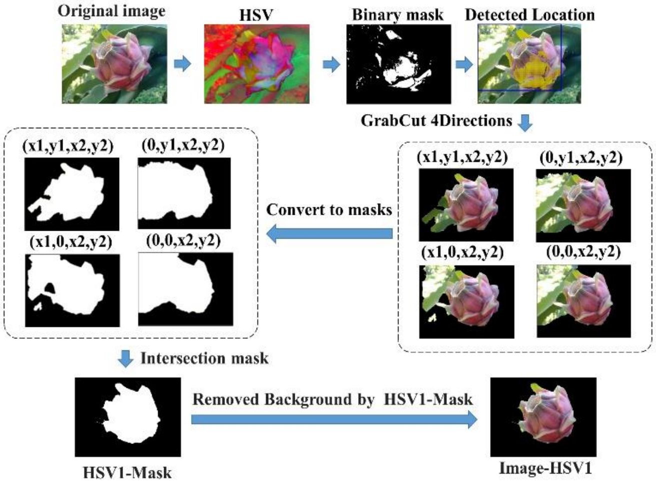


Fig. 6. Illustration of DIP-CBML pre-processing.

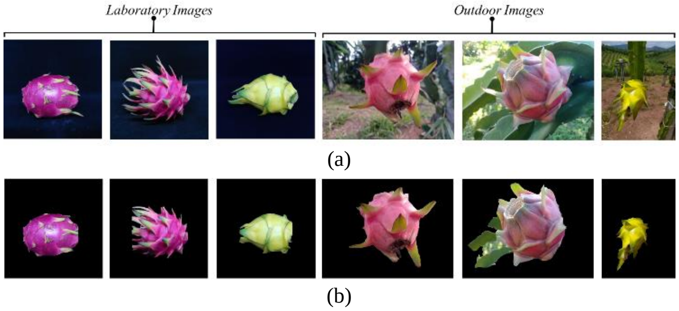


Fig. 7. Examples of output with DIP-CBML pre-processing (a) Original images (b) Image-HSV1.

D. Red and Yellow Classification

Red and yellow classification is the procedure to classify the dragon fruit group into two groups using the characteristics of its peel color, consisting of dragon fruit red peel, and yellow peel. This process has already been discussed in [8]. The laboratory images can use the original image for processing. The results of the laboratory images are highly satisfactory, but the results of outdoor images are not. Therefore, in this research, the Image-HSV1 was used as an input image to classify the dragon fruit groups based on peel color instead of the original image.

The process uses HSV color images to calculate the red ratio and yellow ratio by using the HSV range of red and yellow as the threshold value for counting the number of red pixels and the number of yellow pixels. There is an H value from 23-37 for yellow. The red is divided into two ranges: 0-23 and 151-179. The Saturation (S) and Value (V) values are set from 100-255. Then divide each value by the image size and multiply by 100 to calculate the percentage as Eq. (2) and (3). If any image has a yellow ratio greater than a red ratio, it is an image of yellow peel. The method can immediately be classifying the species of dragon fruit in the image because there is only one species of yellow peel in the dragon fruit.

$$Yellow\ ratio = YP/(image\ size)*100 \quad (2)$$

$$Red\ ratio = RP/(image\ size)*100 \quad (3)$$

where YP is the number of yellow pixels and RP is the number of red pixels.

This work has performed greatly for the results of outdoor images. For laboratory images, the results are still satisfactory. This step reduces the volume of the dataset because the remaining data are red peel groups and affects processing time. It is an advantage of adding this step into research operations.

E. Image Background Removal

Image background removal is a process in which only the background is removed from the image. From Fig. 7(b), the pre-process can remove the background of some images completely, especially the laboratory images. While some outdoor images still have backgrounds left. Therefore, DIP-CBML added this process to remove the remaining background, especially the outdoor image by following the steps in algorithm 3. The illustration of image background removal process is shown in Fig. 8.

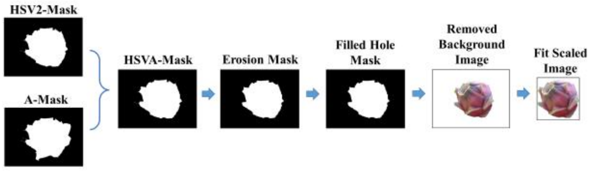


Fig. 8. Illustration of image background removal.

Algorithm 3: Image Background Removal

Input: Original image with RGB color model.

Output: Fit scaled image.

HSV2-Mask = Result of remaining background removal by HSV color model.

A-Mask = Result of background removal by Lab color model.

HSVA-Mask = Result of Intersection A-Mask and HSV2-Mask

Erosion Mask = Result of Erosion HSVA-Mask with kernel 5x5

Filled Hole Mask = Result of Fill Hole HSVA-Mask by OpenCV Library

Removed Background Image = Result of Removing Background in the input image by Filled Hole Mask

Fit Scaled image = Result of cropping proportion of fruit and fit scale its size not over 100 on each side

This process is designed with two background removals, first using an output of pre-process procedure as shown in algorithm 2; input to a subprogram of the remaining background removal by the HSV color model and the second using the original image input to subprogram of the background removal by lab color model. Both subprograms yield two masks named HSV2-Mask as Fig. 10(b) and named A-Mask as Fig. 10(c) respectively. After that take the two masks to the intersection represented by the HSVA-Mask as Fig. 10(d). Use the morphology function erosion to remove extraneous pixels in the HSVA-Mask with kernel size 5x5 and fill the hole with the OpenCV library. Then use the modified HSVA-Mask to remove the background from the image where black pixels refer to background pixels and white pixels refer to dragon fruit pixels. Any coordinate in the original RGB image with the same coordinate as the background pixel will be changed to 255. The result will be a dragon fruit image with a white background, as Fig. 10(e), and the final step is to fit scale the image of the dragon fruit proportion to 100x100 pixels, the result is Fig. 10(f).

1) Subprogram of remaining background removal by HSV color model: This subprogram is the remaining background removal process. The input data is the result of pre-processing as shown in algorithm 2 and Fig. 6. consisting of Image-HSV1, HSV1-Mask, and Coordinates (x1,y1,x2,y2). This data will be used as initial processing steps in the algorithm 4 which is processed similarly to pre-processing as mentioned in Section 3(C). The output of the pre-processing procedure was reprocessed again to remove any pixel that expected part of the background out of the image. The final output of this subprogram is a binary image named HSV2-Mask that can be used to split the background off.

As algorithm 4, starting from the processing of the check mask and image subprogram as shown in algorithm 5, the Image-HSV1 is recalculated for the yellow ratio and the HSV1-Mask is detected for the new coordinates, represented by coordinates (a1,b1,a2,b2) by the detect location of dragon fruit subprogram as Fig. 5 and take the coordinates (a1,b1,a2,b2) to remove the background from the image by 4 graph cut. The result is a binary image represented by the RM-Mask.

If the background has been completely removed, the yellow ratio of Image-HSV1 will be equal to 0 that means not a yellow pixel in the image already or the new coordinates (a1,b1,a2,b2) must be equal to the old coordinates (x1,y1,x2,y2) that means if continue to remove background again, the result is still the same, but if these conditions are not met, the RM-Mask will be used as a mask to remove the background from the original image again; where black pixels refer to background pixels and white pixels refer to dragon fruit pixels. Any coordinate in RGB image that has the same coordinates as a background pixel is changed to 0. The result is RGB image represented by Image-HSV2. The yellow ratio of Image-HSV2 is recomputed and increases the value of the NR variable by 1 to count the number of background removal attempts. This will not remove the background more than 3 times.

After that, if the yellow ratio of Image-HSV2 is equal to 0 or the coordinates (a1,b1,a2,b2) are equal to the original coordinates. (x1,y1,x2,y2) sets the HSV2-Mask value to be the same as the RM-Mask and terminates the subprogram immediately because the background is considered eliminated, but if the conditions are not met, it will continue to remove the background by using the RM-Mask as the initial data to remove the next background.

The process is restarted using the RM-Mask as the starting point, creating a new set of inputs: specifying new (x1,y1,x2,y2) coordinates with the detect location of dragon fruit subprogram, creating a new HSV1-Mask by 4 Graph Cut that used new (x1,y1,x2,y2) coordinates and reconstructing new Image-HSV1 with a new HSV1-Mask. After that, the new HSV1-Mask and new Image-HSV1 were processed with the check mask and image subprogram again to compute the yellow ratio, specify the coordinates (a1,b1,a2,b2), and finally create a new RM-Mask. This new RM-Mask will be used to reconstruct the Image- HSV2 and calculate the yellow ratio of the new Image-HSV2, with the NR variable increased by 1 as before. The image with the background removed will have a yellow ratio of 0 or coordinates (a1,b1,a2,b2) equal to co-ordinates (x1,y1,x2,y2). If this condition is met, the background removal will be stopped. Set the mask HSV2-Mask as same as RM-Mask. On the other hand, we will remove the background repeatedly, but not more than three times, because it may remove the background too much and cause the dragon fruit part to be removed as well. The results obtained from this step are shown in Fig. 10(b).

Algorithm 4: Remaining Background Removal by HSV Color Model

```
Input: Image-HSV1 , HSV1-Mask and (x1,y1,x2,y2).
Output: HSV2-Mask.
Yellow_Ratio, (a1,b1a2,b2) and RM-Mask = Return values of Check
Mask and Image by Image-HSV1 and HSV1-Mask.

NR = 0 /*stand for number of removing*/
If (Yellow_Ratio == 0)  then
    RM-Mask = HSV1-Mask
else
    If ((a1,b1a2,b2)==(x1,y1,x2,y2)) then
        RM-Mask = HSV1-Mask
        Yellow_Ratio = 0
    else
        Image-HSV2 = Remove Background by RM-Mask
        Yellow Ratio = Yellow Ratio of Image-HSV2
        NR++
    End
End
While (Yellow_Ratio!=0 && (a1,b1a2,b2)!=(x1,y1,x2,y2)) do
    If (NR<=3) then
        (x1,y1,x2,y2) = Detecting Dragon Fruit Location by RM-Mask
        While ((x1,y1,x2,y2) == Border of image) do
            Erosion RM-Mask with kernel 5x5
            (x1,y1,x2,y2) = Return values of Detecting Dragon Fruit
                            Location by erosioned RM-Mask
        End
        HSV1-Mask = GrabCut 4Directions by (x1,y1,x2,y2)
        Image-HSV1 = Remove by HSV1-Mask
        RM-Mask = Return values of Check Mask and Image by
                  Image-HSV1 and HSV1-Mask.

        Image-HSV2 = Remove background by RM-Mask
        Yellow_Ratio = Yellow Ratio of Image-HSV2
        NR++
    else
        Yellow_Ratio = 0
    End
End
HSV2-Mask =RM-Mask
```

Algorithm 5: Check Mask and Image Subprogram

```
Input: image and mask data.
Output: Yellow_Ratio of input image, Coordinates (a1,b1a2,b2) and
        RM-Mark.
Yellow_Ratio = Yellow Ratio of input image.
(a1,b1,a2,b2) = Detect Location of Dragon Fruit by input mask.
RM-Mark = GrabCut 4Directions  by (a1,b1,a2,b2)
```

2) Subprogram of background removal by lab color model: The subprogram of background removal by lab color model is a process that uses Lab color model's properties to remove background pixels as shown in the algorithm 6. The illustration of background removal by lab color model process as shown in Fig. 9. First, the original image, which is RGB color model, is converted to a Lab color model. After that, a channel is split and converted to a binary image by Otsu' Threshold. Take the binary image into the process of the detect location of dragon fruit subprogram as shown in Fig. 5, which gives the coordinates (x1,y1x2,y2) as the location of the dragon fruit.

As before, take the coordinates (x1,y1,x2,y2) as the rectangle that will be used to remove the background from the image by graph cut with four directions. If (x1,y1,x2,y2) is the same coordinate as the border of the image, Use the morphology function erosion with kernel size 5x5 to modify the binary image before reading the new position again until the coordinates (x1,y1,x2,y2) are not equal to the border of the image.

Take the results from graph cut with four directions and convert them to a mask. Make the dragon fruit part into white pixels. The background part is black. Bring all four masks to find the white pixels together by using the intersection area. The result is a binary image, represented by A-mask which can be used to remove the background from the image. The results obtained from this step are shown in Fig. 10(c).

Algorithm 6: Background Removal by Lab Color Model

```
Input: Original image with RGB color model.
Output: A-Mask.
lab_img = Result of converting input image RGB to Lab color model.
binary_img = Result of converting A channel of lab_img to binary
             image by Otsu’s Threshold

(x1,y1,x2,y2) = Return values of Detecting Dragon Fruit Location by
                binary_img
While ((x1,y1,x2,y2) == Border of image) do
    Erosion RM-Mask with kernel 5x5
    (x1,y1,x2,y2) = Return values of Detecting Dragon Fruit
                    Location by erosioned binary_img
End
IMG1, IMG2, IMG3 and IMG4 = Return values of Grab Cut
4Directions by Coordinates (x1,y1,x2,y2).
M1, M2, M3 and M4 = Return values of Converting IMG1, IMG2,
                    IMG3 and IMG4 to mask
A-Mask = Result of Intersection all Masks (M1, M2, M3 and M4)
```

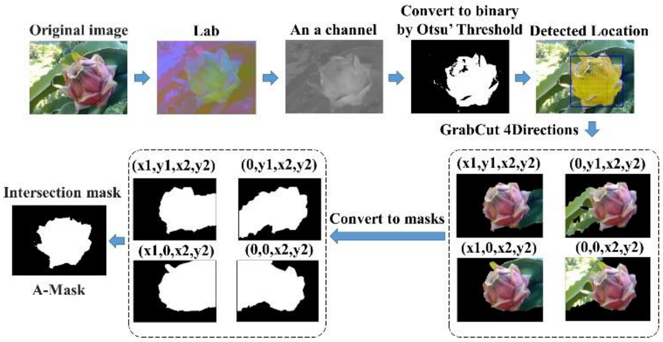
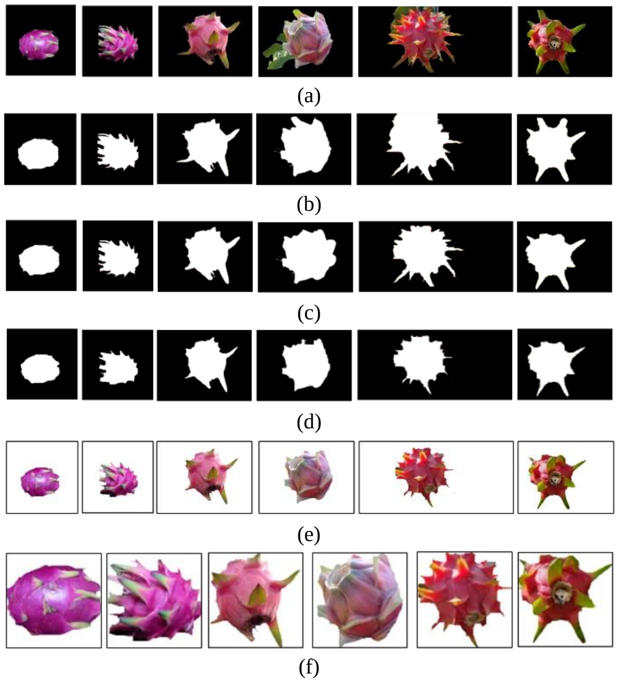


Fig. 9. Illustration of background removal by lab color model subprogram.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 10. Example Output of image background removal (a) HSV1-Image is an input image. (b) HSV2-Mask is an output from subprogram of remaining background removal by HSV color model. (c) A-Mask is an output from subprogram of background removal by Lab color model. (d) HSVA- Mask is an output from intersection A-Mask and HSV2-Mask process. (e) Output from remove background by HSVA-Mask process. and (f) Output from fit scale fruit to 100x100 process.

F. Six Classes of Red Classification

From the procedure in Fig. 1, this method classifies the dragon fruit into two groups from the peel color. Because there is only one species of yellow-skinned in the dataset. For red-skinned dragon fruit, the background in the image will be removed. Then, the dataset will leave only the image of six species of red peel. Therefore, at this stage, only the red-skinned dragon fruit species will be classified by using the content-based technique and deep learning.

Classification using the content-based technique in this research is divided into two parts: feature extraction and classification model. The feature extraction section consists of 24 color attributes, 3 texture attributes, and 2 attributes from edge and line feature. A total of 29 attributes are used as shown in Table I. In The classification model section used a machine learning method by support vector machine algorithm (SVM) with polynomial kernel (degree = 11).

The deep learning models such as DIPDEEP[8], VGG16[38], ResNet50[39], and MobileNetV2[40] were used to classify six species of red peel dragon fruit. All models start with an RGB image size of 100x100 pixels as an input layer and end with six output nodes. There is an optimizer using the Adam algorithm. The learning rate was set at 0.001. For the training process, the batch size and epochs were set at 100. The remaining parameters are not specified using all Keras default values.

For an experiment, DIP-CBML was compared with CBML[9] which is the model that uses content-based techniques, and DIPDEEP[8], VGG16[38], ResNet50[39], and MobileNetV2[40] are the models that use deep learning techniques.

TABLE I. FEATURE EXTRACTION OF DIP-CBML METHOD

Extracted Features	Description	Feature Dimension
Colors	Statistical features (mean, standard deviation, skewness, kurtosis) extracted from color channel R,G,B of RGB color model H of HSV color model and a, b of Lab color model.	24
Texture	Properties features (dissimilarity, contrast and asm) extracted from GLCM of RGB only 0° direction.	3
Edge and Line	The ratio of edge pixels from Canny Edge algorithm and the number of lines from Hough Transform Standard algorithm	2

G. Feature Extraction of DIP-CBML

1) Color feature: Color features used in this research include statistical properties of the RGB, HSV, and Lab color models. All three models were split from three channels to one channel to get six image data: Red (R), Green (G), Blue (B), Hue (H), Red/Green Value (a), and Blue/Yellow Value (b). The pixel data of each image were calculated for the mean, standard deviation, skewness, and kurtosis as Eq. (4) to (7) [41]. The total of color features is 24 attributes.

$$Mean = \frac{1}{N}\sum_{k=0}^{N-1} P_k \tag{4}$$

$$Standard\ Deviation = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}(P_k\text{-}Mean)^2} \tag{5}$$

$$Skewness = \frac{\frac{1}{N}\sum_{k=0}^{N-1}(P_k\text{-}Mean)^3}{\left[\frac{1}{N}\sum_{k=0}^{N-1}(P_k\text{-}Mean)^2\right]^{\frac{3}{2}}} \tag{6}$$

$$Kurtosis = \frac{\frac{1}{N}\sum_{k=0}^{N-1}(P_k\text{-}Mean)^4}{\left[\frac{1}{N}\sum_{k=0}^{N-1}(P_k\text{-}Mean)^2\right]^{2}}\text{-}3 \tag{7}$$

where P_k is the pixel value in order k of image pixel and N is the number of image pixels.

2) Texture feature: Texture features used in this research are the features from the gray-level co-occurrence matrix (GLCM) algorithm of RGB images. For feature extraction, importing RGB images into the GLCM algorithm with the skimage.feature library. Three features of GLCM have been extracted: dissimilarity (D), angular second moment (ASM), and contrast (Ct). Each feature uses horizontal direction information (0°). For a total of three features as Eq. (8) to (10) [42,43].

$$Dissimilarity = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} g(i,j)\cdot|i\text{-}j| \tag{8}$$

$$Angular\ Second\ Moment = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\big(g(i,j)\big)^2 \tag{9}$$

$$Contrast = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(i\text{-}j)^2\cdot g^2(i,j) \tag{10}$$

where n is the number of gray levels, g(i,j) is the value of GLCM normalized where i and j are column and row numbers of GLCM normalized as a square matrix n x n in size. The values of i and j are between 0 and n-1.

3) Edge and line features: Edge and line features are representative of the density of the bracts surrounding the dragon fruit. If there are many bracts, the number of edges and lines will also be large. Each specie of dragon fruit has different a number of bracts. The edge of a fruit part and the direction of a line seem to be characteristics of dragon fruit. The edge and line may be a suitable feature for modeling the dragon fruit classification. Edge and line features are used in this research include the edge pixel ratio by processing with the canny edge algorithm[44,45] and the number of straight lines by processing the hough transform algorithm[46,47].

The canny edge algorithm [44,45] is a popular edge detection algorithm with multi-steps used to detect any pixel expected that is the edge of the object in the image. It consists of the noise removal with a 5x5 Gaussian filter, finding the intensity gradient of the image and direction, removing any unwanted pixels by a local maximum in its neighborhood in the direction of gradient and the last step is selecting whether the pixel expected is the edge or not by two threshold values, minVal and maxVal. Any pixels with an intensity gradient more than maxVal are sure to be edges and less than minVal are sure to be non-edges. Any pixels with an intensity gradient between two thresholds are identified as edges or not by their connectivity. If they are connected to the sure-edge pixels, they are decided to be part of the edges. Otherwise, they will be rejected. It is very important to define two thresholds to get the correct result.

Therefore, to get an edge pixel ratio, this method uses the output image of the image background removal process which is the image that was scaled of the dragon fruit proportion to 100x100 pixels as shown in Fig. 10(f) and converted to a grayscale image. Then bring the grayscale image to frequency distribution by histogram. Set the data layer equal to 10. Use the lower boundary of the first layer and the last layer as lower threshold as minVal and upper threshold as maxVal respectively. The thresholds are processed to determine edge pixels using the canny edge algorithm through the cv.Canny function of the OpenCV library[45]. The resulting image is a binary image with white pixels representing the edge pixels. So, the edge pixel ratio is equal to the number of white pixels divided by the image size and times 100, as shown in Eq. (11).

$$\textit{Ratio of Edge} = (\textit{Edge pixels}/\textit{image size})*100 \qquad (11)$$

For the line features, the Hough transform algorithm[46,47] was used. Typically, it is a detection of the shape in the picture; that shape can be represented in a mathematical form such as lines, circles, ellipses, etc. It can detect the shape even if it is broken. This paper focuses on the straight line detection. The line equation in parametric form as shown in Eq. (12).

$$\gamma=x\cos\theta+y\sin\theta \qquad (12)$$

where x,y is a pixel coordinate, γ (rho) is the perpendicular distance from the origin to the line, and θ (theta) is the angle by this perpendicular line and horizontal axis measured counter clockwise.

That means any pixels that are on the same line, have the same rho and theta value. The first step of the algorithm is creating a 2D array as the accumulator, to collect the calculated values of each pair of rho and theta for each pixel. The array initial with a zero value. Let rows equal to the rho and columns equal to the theta. Any pixel was expected to be an edge of the object in the image, the rho was computed by the pixel coordinate (x,y) and every possible theta. Then increment the values by 1 in the accumulator at the column and row that correspond to the rho and theta value that is a result of computing. The number in each cell of the accumulator means the number of pixels that can be on the same line. In the last, this algorithm required minimum voting as a threshold to decide the group of pixels that can be the line. It may be the minimum length of the line that should be detected.

Therefore, to get the number of straight lines, this method takes the results of the canny edge algorithm to be processed, using the function cv.HoughLines of the OpenCV library[47], setting rho which is the distance resolution of the accumulator in pixels to 1, and theta which is the angle resolution of the accumulator in radians to pi/180. An accumulator value was distributed to the histogram frequency, set the data layer equal to 11, take the data from layers two to six which are normal curve area, calculate the mean that rounded up to make an integer, and set it as the threshold as a minimum length of the line that should be detected. Finally, get the number of straight lines from the result of the cv.HoughLines function. From the hypothesis, any dragon fruit with more bracts around the fruit will also have more straight lines.

4) MinMaxScaler: Because the result of the feature extraction mentioned above gives results in numbers with different scales. Therefore, this method adjusts the scale to the same scaling by MinMaxScaler which will have a scale between 0 to 1 by using the formula as in Eq. (13).

$$\textit{MinMaxScaler Transform}= (x\text{-}min)/(max\text{-}min) \qquad (13)$$

where x is the data, min is the lowest value in the column, and max is the maximum value of the data in the column.

IV. Experimental Results

The image dataset consists of 9,754 laboratory images and 9,067 outdoor images. Each dataset includes seven species of Thai Dragon Fruit. The laboratory images were divided into Jumbo White 1,172 images, Vietnamese White 1,190 images, Pink 1,309 images, Siam Red 1,869 images, Taiwan Red 1,184 images, Ruby Red 1,110 images, and Israel Yellow 1,920 images. The outdoor images were divided into Jumbo White 1,241 images, Vietnamese White 2,478 images, Pink 1,013 images, Siam Red 950 images, Taiwan Red 1,021 images, Ruby Red 871images, and Israel Yellow 1,493 images.

The experiment was divided into three experiments. The first experiment is the image type identification. The second experiment is the red and yellow classification. The final experiment is the classification of six classes from the red peel group.

A. Experiment 1

The first experiment is the image type identification which is laboratory image or outdoor image. Before doing the experiment, images of two types were randomly selected 700 images per type (100 images per species) to study an average of background pixels in a gray level range. Test the range with all data. The experiment results are shown in Table II.

TABLE II. RESULTS OF IMAGE TYPE IDENTIFICATION

Types	Number of Samples	Accuracy (%)	Number of Samples	Accuracy(%)
Laboratory Image	700	100.00	9,754	100.00
Outdoor Image	700	100.00	9,067	100.00
Laboratory Image and Outdoor Image	1,400	100.00	18,821	100.00

B. Experiment 2

The Second experiment is the classification of the dragon fruit group from the peel color which is red or yellow. All images were divided into three methods. There are non-pre-processing, graph cut with [23] localization and pre-processing of DIP-CBML. The experiment results are shown in Table III.

TABLE III. CLASSIFICATION RESULTS BETWEEN RED AND YELLOW WITH THREE DIFFERENT METHODS

Types	Number of Samples	Accuracy(%)		
		Non-pre-processing	*[23]+Graph Cut*	*Pre-processing of DIP-CBML*
Laboratory Image	9,754	100.00	100.00	100.00
Outdoor Image	9,067	97.06	99.96	100.00
Laboratory Image and Outdoor Image	18,821	98.53	99.98	100.00

C. Experiment 3

The Third experiment is the classification of six classes from the red peel group by content-based and deep-learning models. The datasets used for this experiment are the results of the image background removal procedure, which is a dragon fruit image with a white background as Fig. 10(f). Before doing the experiment, the images were randomly selected at 1,000 per class. Any class with less number than 1,000 will be added to the image data by the augmentation technique. Here, this method uses only the horizontal flip image method because the required amount is only small. After that, create the dataset into four forms. The first form is only the 6,000 laboratory images in the dataset which were pre-processed by the creating MKR mask method, namely the Laboratory dataset. The second form is only the 6,000 outdoor images in the dataset which were pre-processed by the DIP-CBML method, namely the Outdoor dataset. The third form combined 12,000 images between the first form and the second form, namely the Mixdata1 dataset. The last form is combined images both the 6,000 laboratory images and the 6,000 outdoor images which are pre-processed by the DIP-CBML method, namely the Mixdata2 dataset. Each dataset was divided into 80%, 10%, and 10% for the training set, the validate set, and the test set, respectively. Using the 10-folds validation technique the DIP-CBML was compared with CBML[9], DIPDEEP[8], VGG16[38], ResNet50[39], and MobileNetV2[40]. The experiment results are shown in Table IV.

V. DISCUSSION

This research presents the classification of Thai dragon fruit species from images that support two types of image datasets; laboratory and outdoor called DIP-CBML. The first experiment results showed that DIP-CBML can identify image types correctly with accuracy 100.00%. Therefore, the proposed method can support two types of images.

After identifying image types, the next process is the classification of the dragon fruit groups; red peel color and yellow peel color. The experiment results showed that the pre-processing of DIP-CBML was able to increase efficiency in classifying dragon fruit groups from peel color by 100% when compared to classifying by the original image. For graph cut with [23] localization, it was found that the dragon fruit group could be classified by the color of the peel with an accuracy of 99.98%. Therefore, the pre-processing of the DIP-CBML gives better results. The volume of the dataset was reduced because the process can stop when a yellow peel specie was founded, the remaining data are red peel groups which affect processing time.

Finally, dragon fruit images were changed the background to white and fit proportion of fruit size not over 100 on each side which were used to perform datasets for the classification of six classes from the red peel group. The experiment results showed that the DIP-CBML can get 100% accuracy for training set, but the other models do not. The DIP-CBML can classify six species of red peel group with 97.85%, 94.00%, 95.53% and 95.05% accuracy for Laboratory dataset, Outdoor dataset, Mixdata1 dataset and Mixdata2 dataset respectively. The confusion matrix of best accuracy which is Mixdata1 dataset can be shown in Fig. 11. and an example of the misclassified images is shown in Fig. 12.

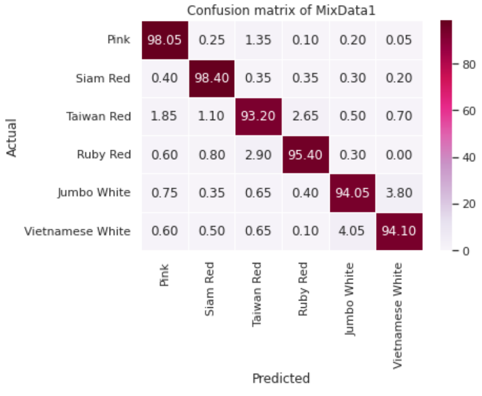


Fig. 11. Confusion matrix of DIP-CBML classification method with Mixdata1 dataset.

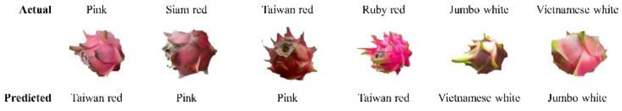


Fig. 12. Examples of the misclassified images.

TABLE IV. COMPARISON FOR ACCURACY OF DIFFERENT CLASSIFICATION METHODS AND DIFFERENT DATASETS FOR 6 SPECIES OF RED PEEL RESULTS BY 10-FOLD VALIDATION

Method	**Mean (SD.)**											
	Laboratory			***Outdoor***			***Mixdata1***			***Mixdata2***		
	Train	***Valid***	***Test***	***Train***	***Valid***	***Test***	***Train***	***Valid***	***Test***	***Train***	***Valid***	***Test***
CBML[9]	100.00 (0.00)	98.28 (0.42)	98.47 (0.62)	99.83 (0.03)	95.17 (1.09)	94.82 (1.24)	99.90 (0.01)	96.54 (0.45)	96.38 (0.68)	99.91 (0.01)	95.98 (0.91)	95.88 (0.56)
DIPDEEP[8]	100.00 (0.00)	98.45 (0.62)	98.80 (0.48)	99.80 (0.05)	94.42 (0.72)	94.18 (1.25)	99.68 (0.31)	95.02 (0.95)	94.96 (1.08)	99.66 (0.39)	94.18 (1.40)	94.34 (0.93)
VGG16[38]	100.00 (0.00)	98.80 (0.32)	98.70 (0.41)	99.81 (0.14)	93.55 (1.46)	93.33 (1.70)	99.78 (0.32)	95.32 (0.81)	95.35 (0.94)	99.69 (0.57)	94.16 (1.40)	94.47 (1.23)
ResNet50[39]	99.73 (0.85)	99.13 (1.64)	99.13 (1.65)	93.27 (18.60)	89.18 (18.15)	89.70 (18.00)	98.37 (3.20)	95.58 (3.55)	95.98 (3.74)	95.26 (9.80)	92.44 (9.97)	92.42 (10.58)
MobileNetV2[40]	86.56 (11.86)	85.43 (12.36)	86.05 (11.85)	73.56 (16.65)	71.35 (16.75)	71.27 (15.63)	90.95 (5.30)	88.53 (5.51)	88.98 (5.58)	70.44 (22.63)	68.88 (21.86)	68.63 (21.94)
DIP-CBML	100.00 (0.00)	97.58 (0.79)	97.85 (0.92)	100.00 (0.00)	94.27 (0.67)	94.00 (0.97)	100.00 (0.00)	95.32 (0.56)	95.53 (0.50)	100.00 (0.00)	95.00 (0.72)	95.05 (0.72)

VI. CONCLUSION

For this paper, the characteristics of laboratory and outdoor images were used to create the classification of Thai dragon fruit species from images that support two image datasets. The DIP-CBML method is used in the completed work. The channel R of RGB color model and gray image were used to identify the type of images between laboratory and outdoor images. The DIP-CBML can identify image type correctly with an accuracy of 100.00% with all datasets. The HSV color model and graph cut algorithm were used to pre-process the images which are able to increase efficiency in classification of dragon fruit groups by peel color with 100% accuracy for outdoor images. As a significant result, the method can classify image group of the laboratory type correctly with 100% accuracy same as other methods. Especially, non-pre-processed gives the same result. Therefore, if the laboratory type was detected, the method can use the original image of laboratory image to classify image groups. The HSV color model, channel of Lab color model and graph cut algorithm were used to remove the background of image which changed the background into white and fit scale on dragon fruit part to 100x100 pixels. The DIP-CBML used these images to classify the six species of Thai red dragon fruit by 29 features from color, texture, edge and line feature. The method gives 95.53% accuracy for both image types with a different pre-processed method. Finally, the entire research process was completed with all components. The precision of each step influences the others. The DIP-CBML may be developed again to increase the accuracy of the last step. The DIP-CBML was designed for the classification, especially Thai dragon fruit species. For other fruits, the DIP-CBML may work or not. This is a challenge for future research. Hopefully, the proposed model will be a guideline for developers to develop tools that can classify species of Thai dragon fruit from images that support pre- and post- harvesting image datasets. This research will add value to the yield of Thai dragon fruit cultivation and can be applied to industrial applications harvesting with robots.

VII. SUGGESTION

After a laboratory image was detected, the DIPDEEP[8] method may be used in the final step because it gives an accuracy of 98.80%. After an outdoor image was detected, the purpose method was used with an accuracy of 95.53%. The combination of several methods in one application is a good approach that works.

REFERENCES

[1] R. Goenaga, A. Marrero, and D. Perez, “Yield and Fruit Quality Traits of Dragon Fruit Cultivars Grown in Puerto Rico,” Horttechnology. United States, vol. 30, no. 6, pp. 803-808, October 2020. https://doi.org/10.21273/horttech04699-20.

[2] T. Perween, K. Mandal, and M. Hasan, “Dragon fruit: An exotic super future fruit of India,” J. Pharmacogn Phytochem. India, vol. 7, no. 2, pp. 1022-1026, Febuary 2018.

[3] A. Trivellini, M. Lucchesini, A. Ferrante, D. Massa, M. Orlando, L. Incrocci, and A. Mensuali-Sodi, “Pitaya, an Attractive Alternative Crop for Mediterranean Region,” Agronomy. Switzerland, vol. 10, no. 8, 1065, July 2020. https://doi.org/10.3390/agronomy10081065.

[4] S. Wichienchot, M. Jatupornpipat, and R.A. Rastall, “Oligosaccharides of pitaya (dragon fruit) flesh and their prebiotic properties,” Food Chem. United Kingdom, vol. 120, no. 3, pp. 850-857, November 2009. https://doi.org/10.1016/j.foodchem.2009.11.026.

[5] S. Kosiyachinda, “Dragon Fruit,” in Thai Youth Encyclopedia, vol. 38, THAI JUNIOR ENCYCLOPEDIA FOUNDATION by His Majesty King Bhumibol Adulyadej The Great. Bangkok: Thailand, 2013, pp. 106-139.

[6] National Bureau of Agricultural Commodity and Food Standards, Ministry of Agriculture and Cooperatives. Thai agricultural standard: Dragon fruit. Available online: https://www.acfs.go.th/standard/download/eng/DRAGON-FRUIT-ENG.pdf (accessed on 9 March 2023).

[7] Department of Agricultural Extension, Ministry of Agriculture and Cooperatives. Agricultural production information service system. Available online: https://production.doae.go.th/service/site/login (accessed on 9 March 2023).

[8] N. Yusamran, and N. Hiransakolwong, “DIPDEEP: Classification for Thai dragon fruit,” Eng. Appl. Sci. Res. Thailand, vol. 49, no. 4 pp. 521–530, Mar 2022.

[9] N. Yusamran, and N. Hiransakolwong, “CBML: Classification Thai Red Dragon Fruit,” ICIC Express Letters, Part B (ICIC-ELB). Japan, vol. 13, no. 11 pp. 1165-1175, November 2022. https://doi.org/10.24507/icicelb.13.11.1165.

[10] S. Przybylek, and C. Cena, What is a Color Model? - Uses & Definition. Available online: https://study.com/academy/lesson/what-is-a-color-model-uses-definition.html (accessed on 9 March 2023).

[11] R. Parekh, Fundamentals of image, audio, and video processing using Matlab®: with applications to pattern recognition, 1st ed. Boca Raton, FL: CRC Press (Taylor and Francis group), 2021.

[12] W. Burger, and M. J. Burge, Principles of Digital Image Processing Fundamental Techniques. NY: Springer Undergraduate Topics in Computer Science – UtiCS, 2009.

[13] Opencv Python Tutorials. Changing Colorspaces. Available online: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_colorspaces/py_colorspaces.html (accessed on 9 March 2023).

[14] M. Elgendy, Deep Learning for Vision Systems, 1st ed. Shelter Island, NY: Manning, 2020.

[15] K. Hameed, D. Chai, and R. Alexander, "A comprehensive review of fruit and vegetable classification techniques," Image Vis Comput. United Kingdom, vol. 80, pp. 24-44, December 2018. https://doi.org/10.1016/j.imavis.2018.09.016.

[16] S. Srisuk, Advanced Image Processing. Bangkok, TH: Mahanakorn University of Technology, 2013.

[17] S. K. Abdulateef, and M. D. Salman, "A Comprehensive Review of Image Segmentation Techniques," Iraqi Journal for Electrical and Electronic Engineering. Iraq, vol. 17, no. 2, pp. 166-175, December 2021. https://doi.org/10.37917/ijeee.17.2.18.

[18] O. Marques, Practical image and video processing using MATLAB. Hoboken, NJ: Wiley-IEEE Press, 2011.

[19] N. T. Bani, and S. Fekri-Ershad, "Content-based image retrieval based on combination of texture and colour information extracted in spatial and frequency domains," The Electron. Libr. United Kingdom, vol. 37, no. 4, pp. 650-66, August 2019. https://doi.org/10.1108/EL-03-2019-0067.

[20] S. Jana, S. Basak, and R. Parekh, "Automatic fruit recognition from natural images using color and texture features," In Proc. the 2nd International Conference on 2017 Devices for Integrated Circuit, DevIC 2017, pp. 620-624, March 2017. https://doi.org/10.1109/DEVIC.2017.8074025.

[21] S. Arivazhagan, R. N. Shebiah, S. S. Nidhyanandhan, and L. Ganesan, "Fruit Recognition using Color and Texture Features," J. emerg. trends comput. inf. sci., vol. 1, no. 2, pp. 90-94, October 2010.

[22] G. Muhammad, "Automatic date fruit classification by using local texture descriptors and shape-size features," In Proc. the UKSim-AMSS 8th European Modelling Symposium on Computer Modelling and Simulation, EMS 2014, pp. 174-179, October 2014. https://doi.org/10.1109/ems.2014.63.

[23] L. Fu, J. Duan, X. Zou, G. Lin, S. Song, B. Ji, and Z. Yang, "Banana detection based on color and texture features in the natural environment," Comput. Electron. Agric. Netherlands, vol. 167, 105057, December 2019. https://doi.org/10.1016/j.compag.2019.105057.

[24] J. F. Reyes, E. Contreras, C. Correa, and P. Melin, "Image analysis of real-time classification of cherry fruit from colour features," J. Agric. Eng. Italy, vol. 52, no. 4, pp. 1-6, December 2021. https://doi.org/10.4081/jae.2021.1160.

[25] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications," Proc. IEEE., vol. 109, no. 3, pp. 247 - 278, March 2021. https://doi.org/10.1109/JPROC.2021.3060483.

[26] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Neural Networks. United Kingdom, vol. 61, pp. 85-117, January 2015. https://doi.org/10.1016/j.neunet.2014.09.003.

[27] S. L. Raja, N. Ambika, V. Divya, and T. Kowsalya, "Fruit Classification System Using Computer Vision: A Review," Int. j. trend res. dev. India, vol. 5, no. 1, pp. 22-26, February 2018. https://doi.org/10.31219/osf.io/kt75d.

[28] H. Muresan, and M. Oltean, "Fruit recognition from images using deep learning," Acta Univ. Sapientiae, Inform. Romania, vol. 10, no. 1, pp. 26-42, August 2018. https://doi.org/10.2478/ausi-2018-0002.

[29] R. E. Masithoh, H. Tampani, and L. Sutiarso, "Detection of White Dragon Fruits (Hylocereus Undatus) in Outdoor Environment Using Image Processing Technique," Aust. J. Basic & Appl. Sci. Jordan, vol. 7, no. 9, pp. 254-260, July 2013.

[30] M, Momeny, A. Jahanbakhshi, K. Jafarnezhad, and Y. D. Zhang, "Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach," Postharvest Biol. Technol. Netherlands, vol. 166, 111204, August 2020. https://doi.org/10.1016/j.postharvbio.2020.111204.

[31] Y. Osako, H. Yamane, S. Y. Lin, P. A. Chen, R. Tao, "Cultivar discrimination of litchi fruit images using deep learning," Sci. Hortic. Technol. Netherlands, vol. 269, 109360, July 2020. https://doi.org/10.1016/j.scienta.2020.109360.

[32] H. Cecotti, A. Rivera, M. Farhadloo, and M. A. Pedroza, "Grape detection with convolutional neural networks," Expert Syst. Appl. United Kingdom, vol. 159, 113588, November 2020. https://doi.org/10.1016/j.eswa.2020.113588.

[33] J. Liu, Z. Zhao, W. Jia, and Z. Ji, "DLNet: Accurate segmentation of green fruit in obscured environments," J. King Saud Univ. - Comput. Inf. Sci. Saudi Arabia, vol. 34, no. 9, pp. 7259-7270, October 2022. https://doi.org/10.1016/j.jksuci.2021.09.023.

[34] Y. Wang, J. Lv, L. Xu, Y. Gu, L. Zou, Z. Ma, "A segmentation method for waxberry image under orchard environment," Sci. Hortic. Netherlands, vol. 266, 109309, May 2020. https://doi.org/10.1016/j.scienta.2020.109309.

[35] H. S. Gill, O. I. Khalaf, Y. Alotaibi, S. Alghamdi, and F. Alassery, "Fruit Image Classification Using Deep Learning," Comput. Mater. Contin. United States, vol. 71, no. 1, pp. 5135-5150, January 2022. https://doi.org/10.32604/cmc.2022.022809.

[36] R. Kirk, G. Cielniak, and M. Mangan, "L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks," Sensors. Switzerland, vol. 20, no. 1, 275, January 2020. https://doi.org/10.3390/s20010275.

[37] Q. Sun, X. Chai, Z. Zeng, G. Zhou, and T. Sun, "Noise-tolerant RGB-D feature fusion network for outdoor fruit detection," Comput. Electron. Agric. Netherlands, vol. 198, 107034, July 2022. https://doi.org/10.1016/j.compag.2022.107034.

[38] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," In Proc. the 3rd International Conference on Learning Representations ICLR 2015, pp. 398-406, May 2015.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, pp. 770-778, June 2016.

[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018, pp. 4510-4520, June 2018.

[41] P. Inkeaw, Documents for teaching the course process 204123 Introduction to data science (Introduction to Data Science). Chiang Mai, TH: Document Printing Unit Education Services and Student Quality Development Faculty of Science Chiang Mai University, 2021.

[42] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. Syst. Man Cybern. United States, vol. SMC-3, no. 6, pp. 610-621, November 1973. https://doi.org/10.1109/TSMC.1973.4309314.

[43] Y. Park, and J. M. Guldmann, "Measuring continuous landscape patterns with Gray-Level Co-Occurrence Matrix (GLCM) indices: An alternative to patch metrics?," Ecol. Indic. Netherlands, vol. 109, 105802, February 2020. https://doi.org/10.1016/j.ecolind.2019.105802.

[44] J. Canny, "A Computational Approach to Edge Detection," IEEE Trans. Pattern Anal. Mach. Intell. United States, vol. PAMI-8, no. 6, pp. 679-698, November 1986. https://doi.org/10.1109/TPAMI.1986.4767851.

[45] Opencv Python Tutorials, Canny Edge Detection. Available online: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorial s/py_imgproc/py_canny/py_canny.html (accessed on 9 March 2023).

[46] P. Mukhopadhyay, and B. B. Chaudhuri, "A survey of Hough Transform," Pattern Recognit. United Kingdom, vol. 48, no. 3, pp. 993-1010, March 2015. https://doi.org/10.1016/j.patcog.2014.08.027.

[47] Opencv Python Tutorials, Hough Line Transform. Available online: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html (accessed on 9 March 2023).