Characteristics of Basil in Aspects of Digital Information Retrieval and Data Mining

Varin Chouvatut[1], Ekkarat Boonchieng[2]

Department of Computer Science, Faculty of Science, Chiang Mai University, Thailand
varinchouv@gmail.com, ekkarat@boonchieng.net

Abstract

This research aims to propose a novel process for classifying two types of basil, which have different species but share common genus, based on main characteristics of their leaves. One type of the basil proposed in this research is called holy basil and another type is called sweet basil. Considering their leaves, both types have similar features in various points of view, for example, their color shade, size or dimensions, scribbled patterns of the leaf edge, etc. They have a lot of similar features because they share the same genus. However, their similar features are the challenged characteristics this research aims to explore and also to demonstrate how well our proposed method work.

Keywords: Holy basil, Sweet basil, Classification, Edge detection, Color texture

1 Introduction

There are a certain number of researches [3, 8-10] exploring and classifying various types of plants' leaf recently. However, research with leaf classification for plants sharing common genus but being in different species is still rarely found. The major reason is that any plants sharing common genus will generally have very similar features and thus this causes a big challenge of recognizing each species of theirs. Most researches usually worked on categorizing leaves of different species and also of different genera.

To obtain the segmentation of region of interest (ROI), in this case region of a plant leaf, color image should be converted into gray-scale image and then binary image [4]. Then, the binary segmentation of an individual leaf can be extracted.

Yin et al. [5] showed a number of leaf vein extraction methods which can generally be used for object's edge detection. They compared results obtained from filters including Gabor, Canny, Prewitt, Sobel, etc. From their venation extraction images, we found that Sobel operator provided good details on a plant leaf and this characteristic function could be useful for our purpose of feature extraction.

Once the edge of a plant leaf can be obtained, basic structuring elements [6] of the leaf's edge should be analyzed to extract the direction of edge's Gradient vector [13]. Since Gradient vectors as used in this research of a plant leaf's edge can indicate one of the plant's features.

Other than identification from a leaf's edge shape, green-color shading of a leaf can also be used for a leaf analysis as used in [2]. Singh et al. [2] showed possible green shadings of a plant leaf and they chose HSI color space to analyze green color of a leaf since their experiments may suffer from environmental light condition. In addition, a green leaf may have disease spots on it as shown in [7] and some detection technique must be performed. We thus used hole-filled algorithm [14] to cope with this problem which may occur in any type of a plant leaf.

Vijayashree et al. [1] proposed a method for measuring whether an input plant leaf was a basil leaf. For our research, we choose holy basil and sweet basil as our example of classifying two species of plants whose genus is common. In order to demonstrate our proposed methodology, all information extracted by a number of computer processing techniques together with analysis from some data mining classifiers is combined to achieve the successful classification.

Both holy basil and sweet basil are in common genus called Ocimum; while the holy basil is in Sanctum species, the sweet basil is in Basilicum Linn species. In other words, plant nomenclature of the holy basil is Ocimum Sanctum and plant nomenclature of the sweet basil is Ocimum Basilicum Linn. Since both species of the basil share common genus, they have various visible features and characteristics which are very visually similar in many aspects, discarding their exclusive smell, so that less experiential persons found hard to categorize them apart.

To evaluate how appropriately the characterized features of leaves of each basil species are selected and how well the proposed methods for feature information retrieval work, six classifiers of the support vector machine (SVM) will be used and the accuracy obtained from each classifier is measured. Six SVM classifiers include linear SVM, quadratic SVM, cubic SVM, fine

[*]Corresponding Author: Ekkarat Boonchieng; E-mail: ekkarat@boonchieng.net

DOI: 10.3966/160792642019052003030

Gaussian SVM, medium Gaussian SVM, and coarse Gaussian SVM.

2 Methodology

The following sections explain how to acquire and store a number of dominant characteristic features which can be used for distinguishing the holy basil species and the sweet basil species. To distinguish or categorize the two basil-species, several SVM classification models will be created and then each generated model with the extracted features can be evaluated using the accuracy measurement. Section 3 explains how to extract basil leaves out of a color digital image. Section 4 explains how to retrieve the edge patterns of the basil leaves. Section 5 explains how to observe the color shadings of the basil leaves although every basil leaf looks green. Section 6 shows whether the leaf's dimensions in terms of height and width of each basil-species are different and how to gain the information. And Section 7 shows the results obtained from analyzing the extracted information of the basil leaves using several SVM classification models that are generated. Also, how perfect the generated models work and how perfect the information retrieval in forms of characteristic features is chosen and done.

3 Acquisition of Leaf's Color Image

In this section, we will emphasize on obtaining individual leaf out of a color image with many leaves.

All input images are scanned by a scanner. Using scanned images reduces burden of processing in camera calibration due to the possibly-distinctive skewness of each camera's lens. Also, brightness condition can be controlled and consistent using a scanner to capture input images. Several leaves of holy basil are arranged untidily on a white paper. This process of image acquisition is also applied to leaves of sweet basil.

Figure 1 shows an example image of 34 holy basil leaves of various sizes. One may observe that the leaves are unintentionally arranged on the paper. And varied sizes of the holy basil leaves are mixed together in the image with no arrangement pattern. Thus, some of our input images may have larger number of small leaves than the number of large leaves and some may have larger number of large leaves than the number of small leaves. In every experiment, we did not make any determination of the leaf's size. We just randomly chose leaves from a pile of the holy basil leaves.

Figure 2 below shows an example image composed of 33 sweet basil leaves of different sizes. Again, varied-size leaves are unintentionally arranged on the paper. One may observe that this image has many small leaves of the sweet basil. Although those amongst the small leaves still have variety in their size.

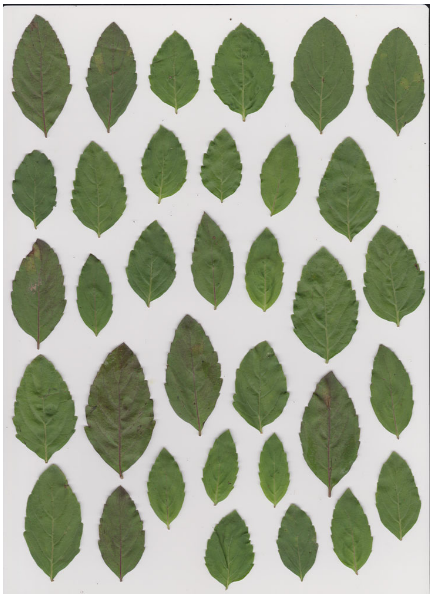


Figure 1. Example of an input image with several leaves of holy basil

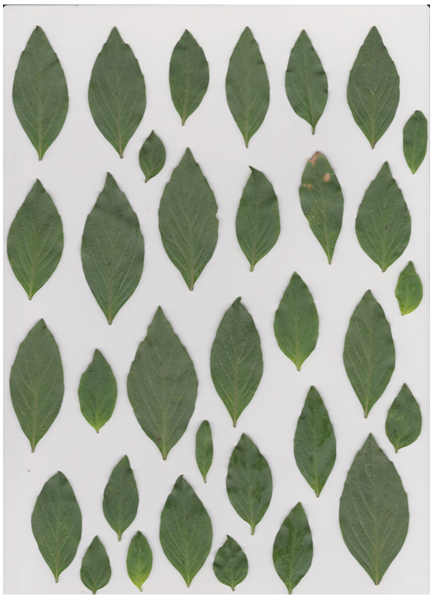


Figure 2. Example of an input image with several leaves of sweet basil

From Figure 1 and Figure 2, we may see that leaves of holy basil and leaves of sweet basil are much similar in many aspects such as size, shape, color, edge pattern, vein, etc.

In terms of size, size of a holy basil leaf and size of a sweet basil leaf are almost the same. As in Figure 1 and Figure 2, holy basil leaves and sweet basil leaves have almost the same width and height, in other words they have almost the same dimensions, regardless of whether they are a small (or young) leaf or a large (or old) leaf.

In terms of shape, leaves of the two species of basil are still similar in shape in the way that the whole leaf's shape is quite like an oval. In details, the leaf's tip is narrow and sharp, its widest portion is about the middle of the leaf but nearer to its bottom, and its bottom has even sharper shape than its tip. Nevertheless, it is a small (or young) leaf or a large (or old) leaf.

In terms of color shadings, the shade of green on a young leaf is lighter than that on an older leaf. Other than the green shadings on the leaf, there are still some degree of pink (or purple) shadings that a basil leaf may have on it. However, a slightly distinct amount of the shadings or even different areas of such distinct shadings on a leaf may be useful in characterizing the basil species.

In aspects of the edge pattern between the two species of basil, having a glance at only one leaf of basil without any comparison (i.e. there is not another species to be compared to it), one can find hard to say the leaf is of holy basil or of sweet basil. This is because both species' leaves have scribbled edge. From visual observation, one may see however not clearly that a holy basil leaf seems to have higher variation of scribbled edge than a sweet basil leaf. However, such difference is not obvious especially in case of small (or young) leaves.

Although leaves of the two species seem to have similarity in various aspects as mentioned above, we are sure that there must be some small degree of distinction in each feature between them. Such minor distinction in characteristic features between the two species of basil should be able to be extracted and thus analyzed deeply to distinguish them apart.

Each of Figure 1 and Figure 2 consists of a large number of leaves of the holy basil and the sweet basil, respectively. Thus, each individual basil leaf must be cropped out of the whole image first. To do so, the background subtraction technique is required.

The foreground or the object of interest for our purpose is the basil leaf detected in the input image. Our input image is an image with several basil leaves. Our background (anything else in the image which is not a basil leaf) has white color since we used a plain A4 white paper to put those basil leaves on it. To extract all foreground objects out of the white background, we can use some segmentation techniques for the purpose. The segmentation techniques we chose are thresholding technique with assistance of edge detection technique.

Firstly, since we do not need to consider the leaf's color shadings at this point, we thus convert the color image into a gray-scale image. The gray-scale image of holy basil leaves is shown in Figure 3. And the gray-scale image of sweet basil leaves is shown in Figure 4.

After that, since there may be some noise in the input image as described in [11-12], for example the brown or black dust on the white background paper, such noise should be removed prior to foreground detection. Median filter has been used for noise reduction, most of the time, it can even eliminate the noise. Median filter is enough to remove such noise in our case and also still keeps much of the detail of the whole image. The filter used in our research has dimensions of 3 × 3 pixels as shown in Figure 5. Since median filter is a nonlinear filter, it has no coefficient to be determined, and thus only its dimensions can be determined. This 3 × 3 median filter has a total of nine elements to be considered. Nine elements of neighborhood pixels of the input image where the filter is overlaid will be considered at a time.

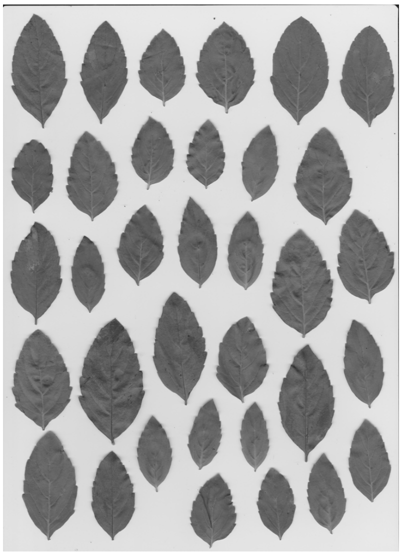


Figure 3. The gray-scale image of holy basil leaves after conversion

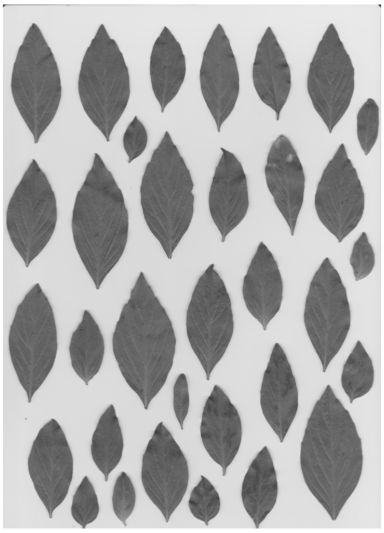


Figure 4. The gray-scale image of sweet basil leaves after conversion

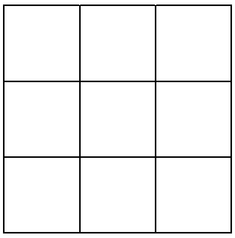


Figure 5. A median filter with dimensions of 3 × 3 pixels used in the research

For example, if the input image is as small in size as 4 × 4 pixels with intensity level of each pixel as shown in Figure 6. Note that, the color input image used in our research has intensity levels in the range of [0, 255] for all components, red, green, and blue. And when the color image is converted into a gray-scale image, its intensity levels will still vary between 0 and 255. Assuming that the 3 × 3 median filter recently overlays the 4 × 4 input-image at the position displayed in gray region and bordered with dash lines in Figure 6. After sorted in ascending order, the list of nine sorted intensity levels will be 0, 5, 8, 10, 39, 97, 100, 150, and 255. So now we can find the median value in the list which is the intensity level 39. Then, this median value of 39 will be updated to the output image from this filtering process at the position as shown in Figure 7.

10	5	97	105
39	8	255	228
0	100	150	176
80	98	148	200

Figure 6. An example of a 4 × 4 gray-scale image with intensity levels in a range of [0, 255] overlaid by a 3 × 3 filter in the region displayed in gray and bordered with dash lines

	39		

Figure 7. The 4 × 4 gray-scale image resulted from an example of filtering process from Figure 4

Since the background in the input image is white, the leaves will be in darker shadings. We thus find the negative image of the filtered image obtained from applying the median filter using (1), where f(x, y) is the output intensity level of pixel at coordinates (x, y) of the negative image and g(x, y) is the input intensity level of the pixel of the input image.

$$f(x, y) = 255 - g(x, y) \quad \textbf{(1)}$$

The resulting negative image of the image of holy basil leaves is shown in Figure 8 while the negative image of sweet basil leaves is shown in Figure 9.

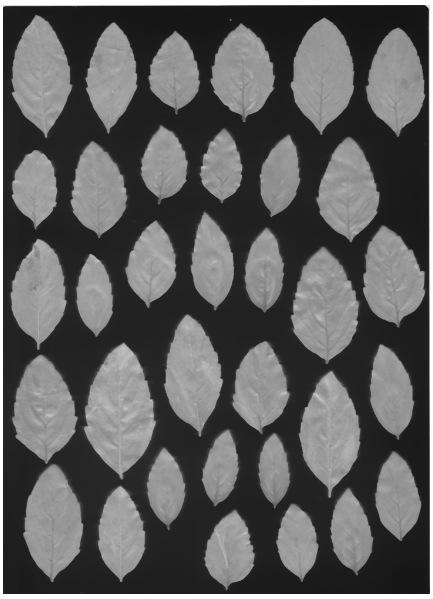


Figure 8. Result of negative image of the image of holy basil leaves

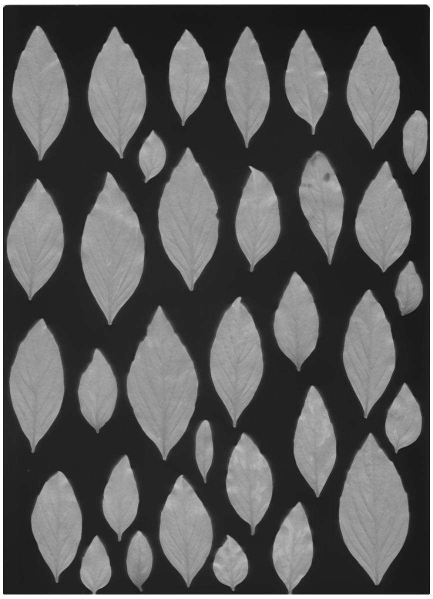


Figure 9. Result of negative image of the image of sweet basil leaves

From Figure 8 and Figure 9, the background of the obtained negative images is changed to a dark tone. Hence, the two negative images can be further processed more easily.

Once we got the negative image with black (or a dark tone) background, we can then use thresholding technique written in (2) to separate each leaf out of the dark background.

$$\begin{aligned} f(x, y) &= 0 \text{ if } g(x, y) < 0.2, \\ f(x, y) &= 1 \text{ if } g(x, y) \geq 0.2. \end{aligned} \quad \textbf{(2)}$$

In (2), the intensity levels are ranged in [0, 1] where level 0 means black color and level 1 means white color. That is, for the intensity ranging in [0, 255], if the input intensity level g(x, y) of the pixel at coordinates (x, y) is less than 51, the output intensity level f(x, y) of the pixel will be changed to 0 (black). But if the input intensity level g(x, y) of the pixel is greater than or equal to 51, the output intensity level will be changed to 255 (white). The intensity level 51 is known as a threshold value used in thresholding technique where the level of 51 is calculated from 0.2 times 255. One may also call the value of 0.2 as the threshold value of (2).

After applying the thresholding technique as explained, the result of image of the holy basil leaves is shown in Figure 10 and the result of image of the sweet basil leaves is shown in Figure 11.

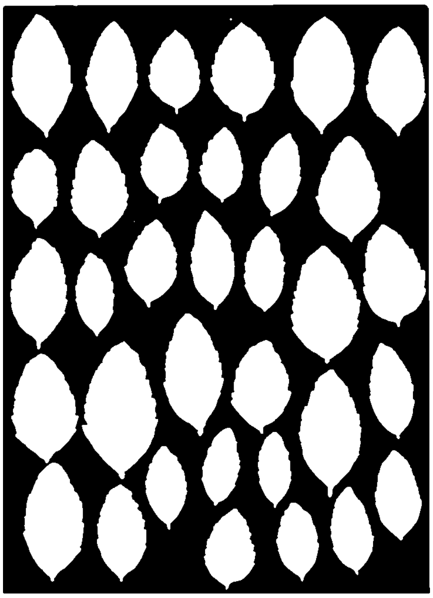


Figure 10. Result of image of holy basil leaves after thresholding

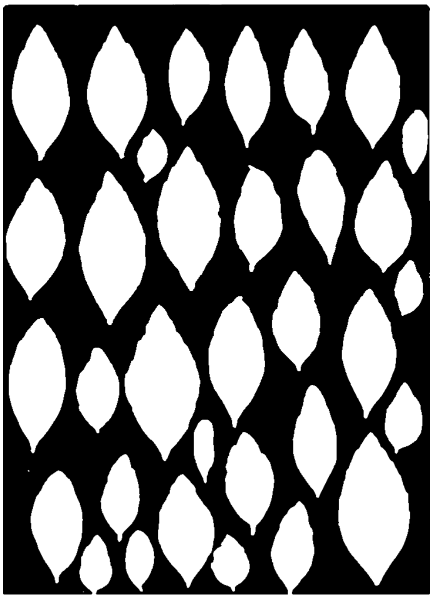


Figure 11. Result of image of sweet basil leaves after thresholding

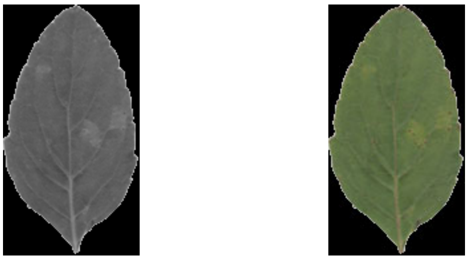
Consequently, each white object in the image resulted from thresholding technique represents region of a basil leaf. Now, an individual leaf can be separated using another technique called blob coloring. The blob coloring technique is to label different objects found in the binary image (the image containing only black and white colors) with different numbers. For example, if an input image has 34 leaves, there will be 34 white objects in the image resulted from thresholding. Thus, there will be objects labelling with numbers 1, 2, …, 34 obtained from blob coloring. The blob coloring technique we used considers 8-neighborhood connection; i.e. if a connected pixel is present in one of eight directions around the current pixel at coordinates (x, y) as shown in Figure 12, the pixel at coordinates (x, y) and the connected pixel are of the same object and thus they two will be assigned with the same label number.

(x–1, y–1)	(x–1, y)	(x–1, y+1)
(x, y–1)	(x, y)	(x, y+1)
(x+1, y–1)	(x+1, y)	(x+1, y+1)

Figure 12. 8-neighborhood pixels considered as connected pixels of the pixel at coordinates (x, y)

After each white blob (or each object) which is each leaf was labelled with a unique number from blob coloring technique, each leaf can be extracted and saved into a separate image file. At this step, there may be a detected object which is not a basil leaf. The object may be blob of a noise region that was not eliminated completely at the filtering step for some reason. However, such blob of noise is obviously small comparing to the blob of a basil leaf. Consequently, we can eliminate or ignore such really small blob out of our interest in the next step.

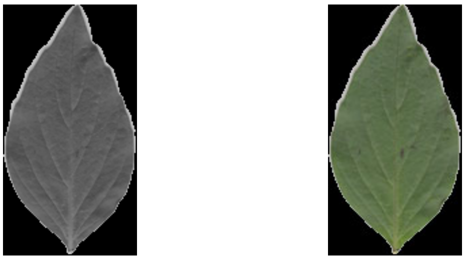
Figure 13 shows an example image file of an individual leaf of holy basil saved separately from applying the blob coloring technique. Figure 13(a) is the gray-scale version of an individual leaf of holy basil and Figure 13(b) is its RGB color version. Note that RGB represents red, green, and blue components of a color image.



(a) and RGB color image (b) of an individual leaf of holy basil

Figure 13. An example of gray-scale image

Also, Figure 14 shows an example image file of an individual leaf of sweet basil saved separately from the blob coloring. Figure 14(a) is the gray-scale version of an individual leaf of sweet basil and Figure 14(b) is its RGB color version.



(a) and RGB color image (b) of an individual leaf of sweet basil

Figure 14. An example of gray-scale image

One may see from Figure 13 and Figure 14 that the boundary of a leaf region saved in an individual image file has no background portion left, in other words, the background region is set to black.

The images of an individual leaf of holy basil and an individual leaf of sweet basil as shown in Figure 13 and Figure 14 are called leaves in a standard position. The standard position chosen in this research is that the highest portion and the widest portion of a leaf are aligned to vertical and horizontal axis, respectively. The standard alignment of a leaf will be necessary for the process of measuring proportion of leaf's dimensions to be described later. If the leaf is not aligned as described, rotation technique calculated from (3) is required. Here, we rotate the object with Δθ = 1° at a time. Note that, positive angle change indicates counterclockwise direction around the object's center point.

$$R=\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(3)}$$

$$P' = R \bullet P.$$

From (3), let P be the current position of a point (or a pixel) of the target object in image with coordinates $[x\ y\ 1]^T$ and P' with coordinates $[x'\ y'\ 1]^T$ be the corresponding point obtained after rotation using the rotation matrix R.

Using (3), the leaf region will be continued rotating until the highest and the widest portions of the leaf is found or at least compromised on the acceptable standard position we defined.

4 Detection of Scribbled Patterns of the Leaf's Edge

Although, a leaf of holy basil and of sweet basil have curved and scribbled edges making them look quite similar, the scribbled patterns of their leaf edge actually have some small degree of difference in details. This difference in patterns can be observed by enlarging the image resulting from edge detection technique shown in Figure 15 and Figure 16. Figure 15 is the enlarged image of an example edge portion of a holy basil leaf. And Figure 16 is that of a sweet basil leaf.

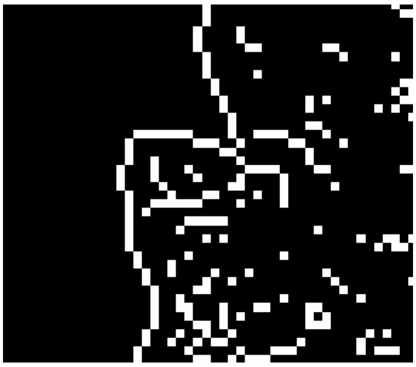


Figure 15. Example of an obtained edge detection showing scribbled patterns of a partial leaf's edge of the holy basil

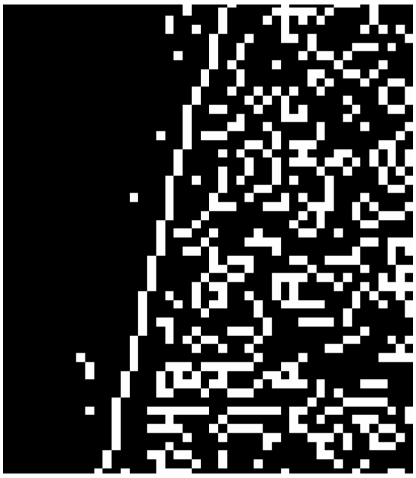


Figure 16. Example of an obtained edge detection showing scribbled patterns of a partial leaf's edge of the sweet basil

To extract patterns of the curved and scribbled edge of a basil leaf, we detected the directions of connected pixels changed using 4-neighborhood pixels as shown in Figure 17.

	(x–1, y)	
(x, y–1)	(x, y)	(x, y+1)
	(x+1, y)	

Figure 17. 4-neighborhood pixels considered as connected pixels of the pixel at coordinates (x, y)

Using 4-neighborhood connection is sufficient to gain characteristics of the scribbled patterns of a leaf. Four directions of neighborhood pixels are used to find the direction or the slope angle of the leaf edge. So, using 4-neighborhood connection of a leaf's edge, horizontal and vertical edge portions are counted to extract the scribbled patterns of the leaf. For the process of edge detection, we used Sobel's edge detection technique [5] which has two operators whose dimensions are 3 × 3 pixels as shown in Figure 18. Figure 18(a) is Sobel's operator for detecting horizontal edges, each of which may be considered as a blob in blob coloring technique. And Figure 18(b) is Sobel's operator for detecting vertical edges.

-1	-2	-1
0	0	0
1	2	1

(a) horizontal edges

-1	0	1
-2	0	2
-1	0	1

(b) vertical edges

Figure 18. Sobel's operators used for the detection of horizontal edges (a) and vertical edges (b)

Before applying the Sobel's operators to the leaf image, the RGB image must first be converted to a gray-scale image and then converted to a binary image which consists of only black and white colors.

Again, thresholding technique is used to convert a gray-scale image to its corresponding binary image. The thresholding technique we used is done using (2) where the threshold value is set to 0.5 for the image's intensity levels varying in the range of [0, 1].

From Figure 13(b), the RGB color image of a holy basil leaf is converted to its corresponding binary image resulting as Figure 19.

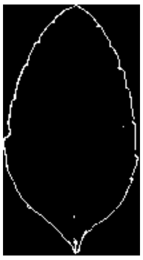


Figure 19. Binary image showing edge detection of a holy basil leaf

Then, edge enhancement technique using one of Sobel's operators, Figure 18(a), to find or emphasize the horizontal edge portions of the holy basil leaf in Figure 19 is applied and results in blob detection of horizontal edges. The detected blobs of horizontal edge portions of the holy basil leaf from Figure 19 is shown in Figure 20. The blob detection is based on the blob coloring technique and thus the number of blobs of the horizontal edges can be counted using this blob coloring technique. In this example, the number of blobs in Figure 20 is 84.

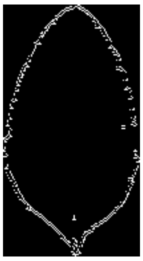


Figure 20. The result of blob detection from horizontal edge enhancement of a holy basil leaf

Other than the horizontal edge portions, the vertical edge portions of the holy basil leaf, Figure 19, are also detected and the detection result is shown in Figure 21. For this purpose, another Sobel's operator, Figure 18(b), is applied to find and emphasize the vertical edge portions of the holy basil leaf in Figure 19. In this example, the number of blobs in Figure 21 is Figure 20.

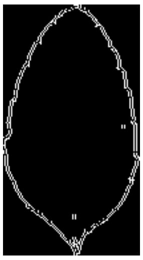
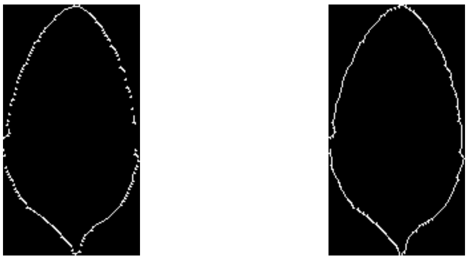


Figure 21. The result of blob detection from vertical edge enhancement of a holy basil leaf

From Figures 20 and Figure 21, one may observe that both images have two small objects inside the boundary of the leaf. The interior objects can be just dark spots on the basil leaf itself or some contaminated noises in the image. We should eliminate such noise out of the leaf region before counting the number of blobs from the horizontal edges or from the vertical edges. To do so, we can use the hole-filled technique once we got the gray-scale image of the basil leaf shown in Figure 13(a).

After applying the hole-filled technique, the detected horizontal and vertical edges in terms of blobs of this holy basil leaf are as shown in Figure 22(a) and Figure 22(b), respectively. From Figure 22, the blobs of noise are completely disappeared. Threshold value from (2) used in this experiment is set to 0.1. The number of blobs counted from Figure 22(a) equals 39 blobs and from Figure 22(b) equals five blobs; note that some blobs may be a long or large connected object thus it is considered to be a single blob.



(a) horizontal (b) vertical

Figure 22. The result of blob detection from horizontal (a) and vertical (b) edge enhancement of a holy basil leaf after hole-filled technique

The hole-filled algorithm can be done in a binary image using the negative technique. Result image obtained from the negative technique is called a negative image. To illustrate how this technique works, let an input binary image whose intensity of each pixel is either 0 (black) or 1 (white) be as shown in Figure 23. Region of the object (e.g. a basil leaf) is shown as white pixels, the black background is the black region outside of the white region, and a hole is shown as a black region inside the white region. Then, we can create a negative image of this binary image using (4) where f_bin(x, y) is the intensity of input image at pixel coordinates (x, y) and f_neg(x, y) is the corresponding intensity of negative image at the same pixel.

$$f\_neg(x, y) = 1 - f\_bin(x, y) \qquad \textbf{(4)}$$

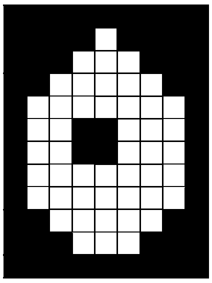


Figure 23. An example of input binary image used for illustration of the hole-filled algorithm

Thus, the obtained result will be as Figure 24.

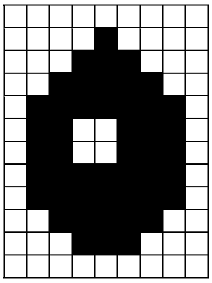


Figure 24. An example of the obtained negative image used for illustration of the hole-filled algorithm

After that within the black region which is now the region of object in the negative image, we can find hole(s) inside the black region more easily using flood-filled algorithm and then replace intensity of any pixels inside the hole's region to black color (color 0 in this case). Consequently, the result image is as shown in Figure 25.

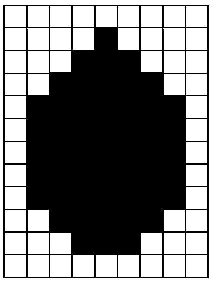


Figure 25. An example of the hole-filled image used for illustration of the hole-filled algorithm

Finally, convert the black and white color back to cancel the effect of creating the negative image at the first step. Again, use (4) to convert the intensity of Figure 25 back, then we will get Figure 26 as a result. Now, the background of the image is black and object (or a leaf's region) is white as same as shown in Figures 10 and 11.

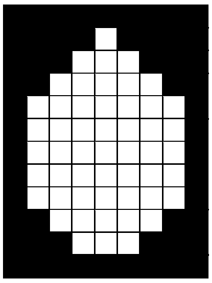


Figure 26. An example of the final image used for illustration of the hole-filled algorithm

From Figure 14(b) and the use of thresholding technique, the RGB color image of a sweet basil leaf is converted to a gray-scale image and then converted to its corresponding binary image resulting as Figure 27.

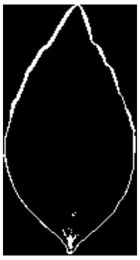


Figure 27. Binary image showing edge detection of a sweet basil leaf

Then, edge enhancement technique using the Sobel's operator from Figure 18(a) is applied to the binary image of the sweet basil leaf shown in Figure 27 and the obtained result of the blob detection of horizontal edges is as Figure 28. In this example, we found 97 blobs of horizontal edges.

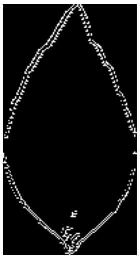


Figure 28. The result of blob detection from horizontal edge enhancement of a sweet basil leaf

Applying another Sobel's operator, Figure 18(b), to the binary image of the sweet basil leaf shown in Figure 27, the detected blobs of vertical edges are as shown in Figure 29. In this example, we found 10 blobs of vertical edges.

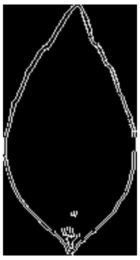


Figure 29. The result of blob detection from vertical edge enhancement of a sweet basil leaf

As same as the case of holy basil leaf, one may see some noises inside the sweet basil leaf in Figure 28 and Figure 29. The hole-filled technique is again applied to the gray-scale image of this sweet basil leaf shown in Figure 14(a). Then, the results of horizontal edges and vertical edges obtained from the hole-filled technique are as shown in Figure 30(a) and Figure 30(b), respectively. The number of blobs counted from the two figures are 43 blobs and 2 blobs, respectively.

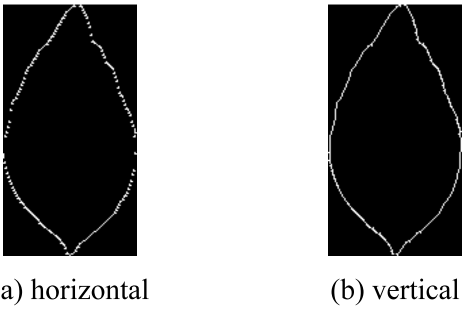


Figure 30. The result of blob detection from horizontal (a) and vertical (b) edge enhancement of a sweet basil leaf after hole-filled technique

Figure 31 shows a comparison of the number of blobs counted from the horizontal edges detected from 143 holy basil leaves and 176 sweet basil leaves. The comparison of the number of blobs counted from the vertical edges detected from both species of basil leaves is shown in Figure 32. The blob detection shown in Figures 31 and 32 is based on the setting of threshold value in (2) to 0.5.

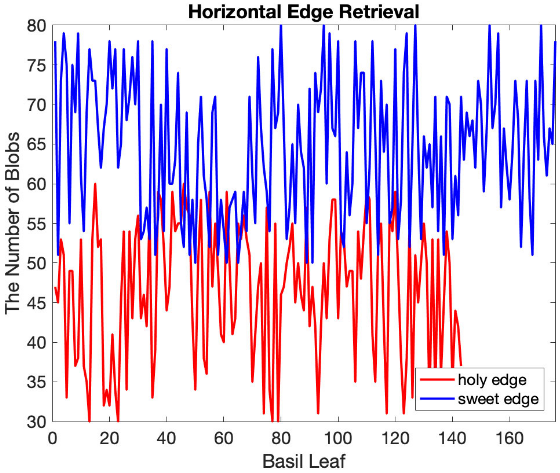


Figure 31. The comparison result of blob detection from horizontal edge enhancement between holy basil leaves and sweet basil leaves

The connectivity information, in the directions of horizontal axis and vertical axis, retrieved above is stored in a tabular format partially shown in Table 1. The connectivity information in terms of blobs of horizontal edges together with blobs in vertical edges is labelled as either 'holy' or 'sweet' indicating basil species, i.e. a holy basil or a sweet basil, respectively.

Table 1. Partial example of information retrieval on connectivity

Horizontal Edges	Vertical Edges	Basil Species
47	1	holy
45	3	holy
53	3	holy
51	2	holy
33	2	holy
:	:	:
78	5	sweet
51	3	sweet
73	6	sweet
79	4	sweet
75	5	sweet

Figure 31, Figure 32, and Table 1 show some degree of distinction between two basil species. Thus, this retrieved information will then be useful for further analysis.

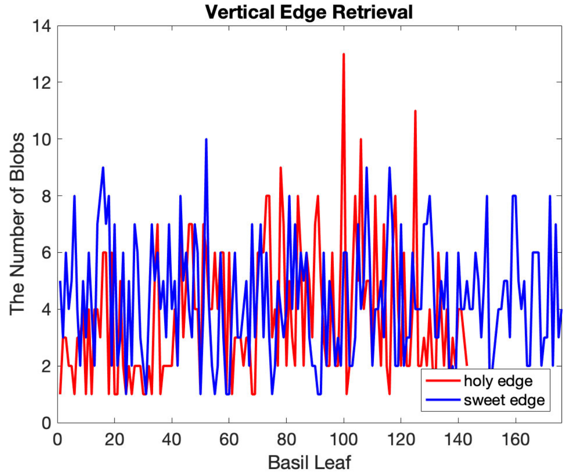


Figure 32. The comparison result of blob detection from vertical edge enhancement between holy basil leaves and sweet basil leaves

5 Separating Uncertain Color Shade of the Leaf

From Figure 1 and Figure 2 of the input color image of holy basil leaves and sweet basil leaves, respectively, the green shadings of the holy basil leaves seem to be lighter than that of the sweet basil leaves. We thus should count the color components of the basil leaves, one of the important classification factors.

Red, green, and blue components of each pixel in the boundary of an example basil leaf are extracted and the color information retrieval is shown in Figure 33. In total, we used 143 holy basil leaves and 176 sweet basil leaves and each of the three color-components has intensity level in a range of [0, 255].

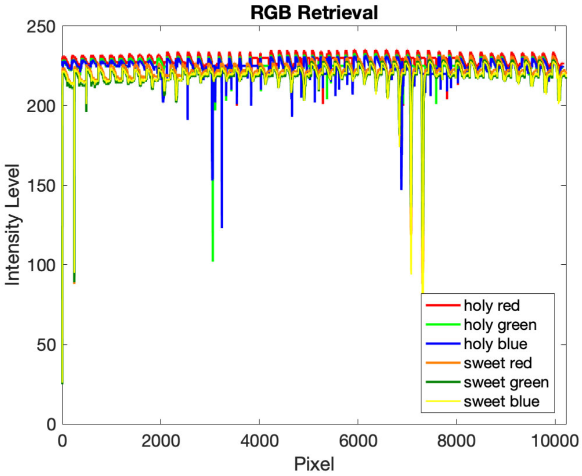


Figure 33. Color information retrieval of a holy basil leaf and a sweet basil leaf

From Figure 33, all three color-components consisting of red, green, and blue component of the example leaf of holy basil mostly have higher intensity levels than the three components of sweet basil.

Here, we stored the average and standard deviation of color shading for all three components of each basil leaf. The average value of each color-component, i.e. red, green, or blue, of the jth basil leaf whose number of pixels equals N_j can be computed using (5). From (5), let $AvgR_j$, $AvgG_j$, and $AvgB_j$ be the average value of red, green, and blue component, respectively, of the jth basil leaf; R_i, G_i, and B_i be the red, green, and blue level, respectively, of the ith pixel on this jth basil leaf; and N_j be the number of pixels of this jth basil leaf; where i = 1, 2, …, N_j and j = 1, 2, …, 143 for holy basil but j = 1, 2, …, 176 for sweet basil.

$$AvgR_j = \frac{\sum_i R_i}{N_j},$$

$$AvgG_j = \frac{\sum_i G_i}{N_j}, \text{ and} \tag{5}$$

$$AvgB_j = \frac{\sum_i B_i}{N_j}.$$

As same as the average calculation, the standard deviation of each color-component of the jth basil leaf whose number of pixels equals N_j can be computed using (6). From (6), let SdR_j, SdG_j, and SdB_j be the standard deviation of red, green, and blue component, respectively, of the jth basil leaf and $AvgR_j$, $AvgG_j$, and $AvgB_j$ be the average value of red, green, and blue component, respectively, obtained from (5).

$$SdR_j = \sqrt{\frac{\sum_i \left(R_i - AvgR_j\right)^2}{N_j}},$$

$$SdG_j = \sqrt{\frac{\sum_i \left(G_i - AvgG_j\right)^2}{N_j}}, \text{ and} \tag{6}$$

$$SdB_j = \sqrt{\frac{\sum_i \left(B_i - AvgB_j\right)^2}{N_j}}.$$

Figure 34 shows a graph plot comparing the average values of red, green, and blue components between two species. And Figure 35 shows a graph plot comparing their standard deviation values.

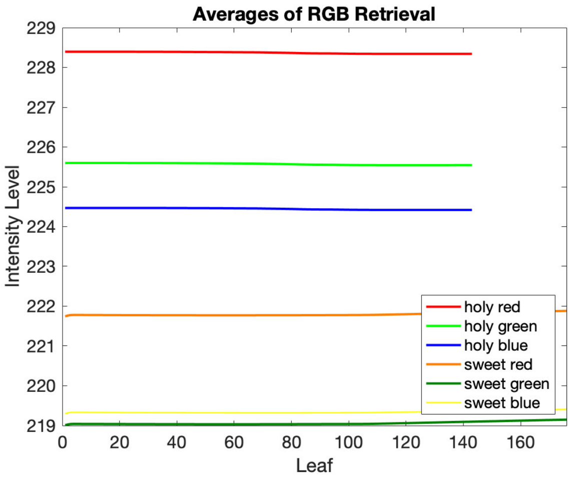


Figure 34. Information retrieval on average color-shadings of all holy basil leaves and all sweet basil leaves

Hence, the average values and standard deviation values of all basil leaves are stored with labels of ‘holy’ or ‘sweet’ indicating holy basil or sweet basil, respectively. Table 2 shows an example portion of the retrieved information in aspect of color-shadings of the basil leaves. Note that the red, green, and blue components are in the range of [0, 255] and the averages and standard deviations are displayed with floating point numbers rounded to two decimal places.

Table 2. Partial example of information retrieval on color-shading

Average			Standard Deviation			Basil
Red	Green	Blue	Red	Green	Blue	Species
228.39	225.60	224.47	4.71	4.64	5.29	holy
228.39	225.60	224.47	4.71	4.64	5.29	holy
228.39	225.60	224.47	4.71	4.64	5.29	holy
228.39	225.60	224.47	4.71	4.64	5.29	holy
228.39	225.60	224.47	4.71	4.64	5.29	holy
:	:	:	:	:	:	:
221.75	219.01	219.30	9.51	8.32	10.80	sweet
221.76	219.03	219.32	9.30	8.09	10.63	sweet
221.78	219.04	219.33	9.22	8.01	10.57	sweet
221.78	219.04	219.33	9.22	8.01	10.57	sweet
221.78	219.04	219.33	9.22	8.00	10.57	sweet

The average values and standard deviation values of each of the three color-components are not obviously different. This is because the overall color shadings of a basil leave, no matter it is of either holy basil or sweet basil, are in quite similar green shadings. However, observing from Figure 34 and Figure 35, the three components of the two basil-species are obviously distinguishable. Thus, the retrieved features of this color shading are stored for further analyzing in the process of model generation.

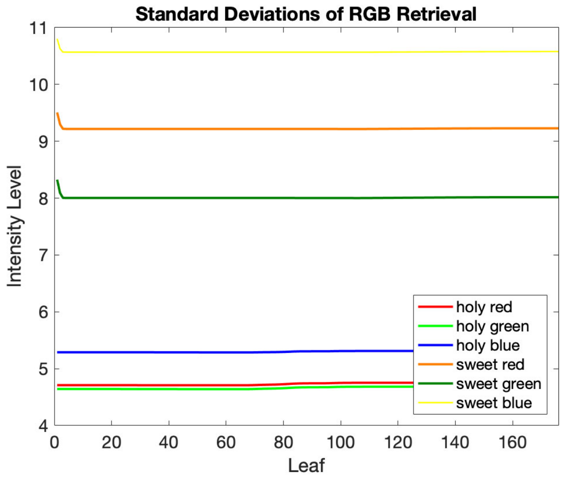


Figure 35. Information retrieval on standard deviations of color-shadings of all holy basil leaves and all sweet basil leaves

6 Proportion of Leaf’s Dimensions

Height of a holy basil leaf seems to be larger than height of a sweet basil leaf while width of a sweet basil leaf seems to be larger than width of a holy basil leaf. Nevertheless, we need some exact measurement in terms of dimensions of each basil leaf. Therefore, this is the reason that we need to rotate (if required) a basil leaf to meet its standard position. A basil leaf may require to be rotated in either counterclockwise direction or clockwise direction until its highest portion aligns to the y axis in the Cartesian coordinate system. Consequently, the widest portion of the leaf, or some size closest to the widest portion of the leaf, will be aligned to the x axis. Note that, we mainly place the highest portion of a basil leaf aligned to the y axis before extracting the widest portion of it. As a result, the highest and the widest portion will be determined as the leaf’s height and width, respectively.

An example of the height and width measured from a holy basil leaf in a standard position is as shown in Figure 36. In this example, height of this leaf is 170 pixels and width is 112 pixels.

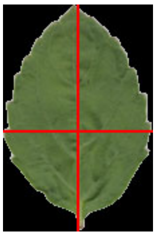


Figure 36. An example of information retrieval on dimensions of a holy basil leaf in a standard position

After the dimensions in terms of height and width in the unit of pixels of a basil leaf can be extracted, the ratio of the two dimensions of the leaf is calculated using (7). From (7), let Ratio$_j$ be the ratio of height,

Height_j, and width, Width_j, of the jth basil leaf where j = 1, 2, …, 143 for the holy basil and j = 1, 2, …, 176 for the sweet basil. Here, we use the ratio of height to width for each leaf because, mostly and naturally, the height of a basil leaf is longer than its width resulting in a ratio value of larger than one. This ratio value of larger than one can help preventing some prone to error in calculation with many precision numbers.

$$\text{Ratio}_j = \frac{\text{Height}_j}{\text{Width}_j} \quad \textbf{(7)}$$

Then, the retrieved information in terms of a leaf's dimensions will be stored together with a label 'holy' or 'sweet'. A partial example of the retrieved information on the basil leaves' dimensions is as shown in Table 3. Note that height and width of each leaf are measured in the unit of pixels.

Table 3. Partial example of information retrieval on Dimensions

Height	Width	Ratio	Basil Species
309	150	2.06	holy
240	105	2.285714286	holy
288	165	1.745454545	holy
252	153	1.647058824	holy
231	151	1.529801325	holy
:	:	:	:
324	198	1.636363636	sweet
309	169	1.828402367	sweet
282	176	1.602272727	sweet
303	164	1.847560976	sweet
318	150	2.12	sweet

From Table 3, one may think that the ratios of two basil-species are really similar or not significantly different, the similarity and difference of the two basil-species in terms of height, width, and ratio measurement can be seen in Figure 37 to Figure 39 below.

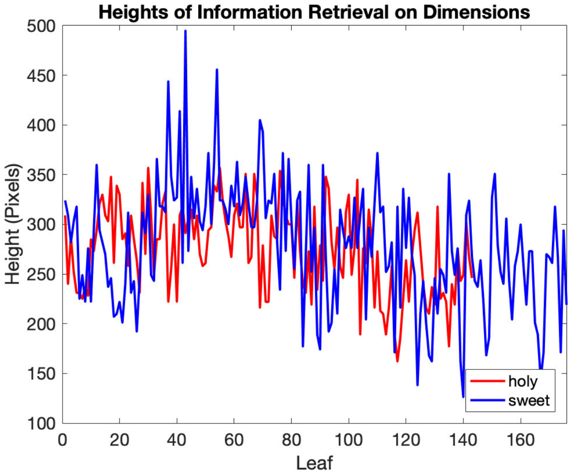


Figure 37. The comparison result of the leaves' heights between holy and sweet basil species

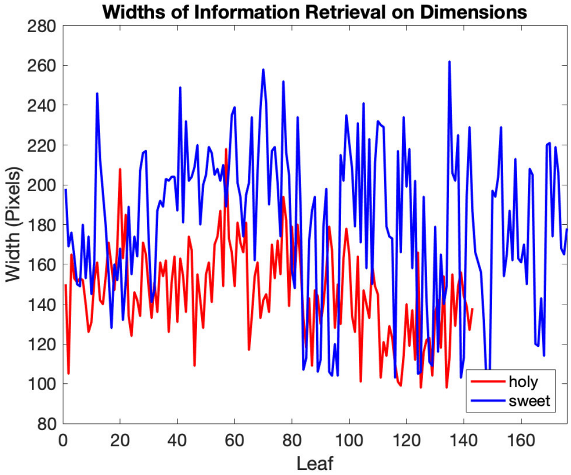


Figure 38. The comparison result of the leaves' widths between holy and sweet basil species

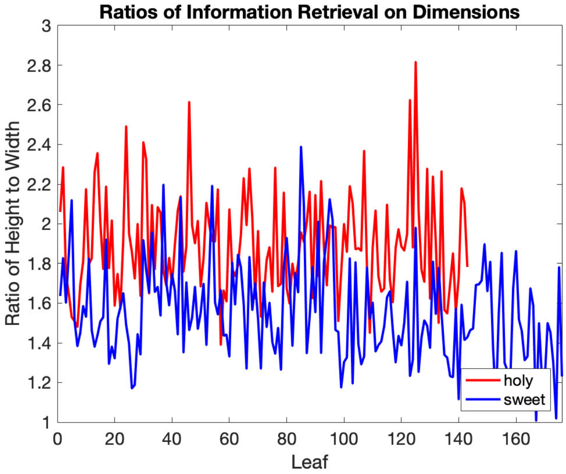


Figure 39. The comparison result of ratios of height to width between holy and sweet basil leaves

From Figure 37, the information on the height dimension of both basil species looks as not different or at least insignificantly different.

From Figure 38, the information on the width dimension of the sweet basil leaves is mostly higher than that of the holy basil leaves.

Although the leaves' heights of both basil species are quite similar, their widths are different. Consequently, the ratios of height to width calculated from (7) between two basil-species are undoubtedly different in some degree as seen in Figure 39. These different characteristics should be able to be used further to categorizing the two species apart.

7 Analyzing the Retrieved Information

To categorize a holy-basil leaf out of a sweet-basil leaf, all dominant features of a leaf of each basil species as mentioned above can assist in generating an efficient classification model. Also, the appropriate feature selection, which should be of dominant features of the basil species, can assist in indirectly reducing or even eliminating the unexpectedly effects of inappropriate or noisy data on the training process of model creation.

Since we have a small number of holy-basil leaves and sweet-basil leaves, i.e. 143 holy-basil leaves and 176 sweet-basil leaves in total, we chose to create various models of the Support Vector Machine (SVM) classifier. A classifier's model can also be utilized for demonstrating our selection of main (or dominant) features of two basil-species and evaluating the performance of our digital information retrieval. In our experiments, we created six models of SVM classifiers consisting of linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian SVM, and coarse Gaussian SVM. Every model is evaluated using the Accuracy measurement calculated from (8):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}, \quad \textbf{(8)}$$

where TP refers to the number of true positive samples, TN refers to the number of true negative samples, P refers to the number of positive samples, and N refers to the number of negative samples.

All of basil leaves we had was split into two sets, one was used for training process called training set and another was used for testing process called testing set. The training dataset is about 80 per cent of the total number of leaves and the testing dataset is thus about 20 per cent. Hence, the basil leaves of each species were divided into two sets as shown in Table 4.

Table 4. The numbers of training and testing dataset determined for each basil species

Species	Training Set	Testing Set	Total
Holy Basil	114	29	143
Sweet Basil	141	35	176

Given only the characteristic features on scribbled patterns of the basil leaves' edges, the accuracy measured from classifying basil species using each of classification models of SVM classifier can be seen in Table 5.

Table 5. Accuracy measurement based on scribbled patterns of the leaves' edges

Measure-ment	SVM Models					
	Linear	Quad.	Cubic	Fine Gaus.	Medium Gaus.	Coarse Gaus.
Accuracy	81.2%	81.5%	80.6%	79.6%	80.3%	81.2%

Given only characteristic features on color shadings of the basil leaves, the accuracy measured from classifying basil species using each classification model of SVM classifier can be seen in Table 6.

Table 6. Accuracy measurement based on color shadings of the leaves

Measure-ment	SVM Models					
	Linear	Quad.	Cubic	Fine Gaus.	Medium Gaus.	Coarse Gaus.
Accuracy	75.9%	52.0%	50.2%	83.4%	82.8%	82.8%

Given only the characteristic features on basil leaves' dimensions, the accuracy measured using each of the classification models of SVM classifier can be seen in Table 7.

Table 7. Accuracy measurement based on the leaves' dimensions

Measure-ment	SVM Models					
	Linear	Quad.	Cubic	Fine Gaus.	Medium Gaus.	Coarse Gaus.
Accuracy	74.7%	81.6%	67.2%	74.1%	80.9%	79.1%

Given all of the characteristic features on basil leaves mentioned above, the accuracy measured from classifying basil species using each of the classification models of SVM classifier can be seen in Table 8.

Table 8. Accuracy measurement based on all characteristic features of the leaves

Measure-ment	SVM Models					
	Linear	Quad.	Cubic	Fine Gaus.	Medium Gaus.	Coarse Gaus.
Accuracy	100.0%	100.0%	100.0%	98.4%	100.0%	100.0%

From Table 5 to Table 8, using only one characteristic feature of the basil leaves is surely not enough for categorizing two species apart from each other efficiently but providing all aspects of characteristic features of basil leaves to train the classification model results in almost perfect model obtained.

From Table 5 given only the scribbled edge patterns of basil leaves, the obtained accuracy measurement is about 80.7 per cent on average. From Table 6 given only the color shadings in terms of average and standard deviation values of three color-components of basil leaves, the obtained accuracy measurement is as low as only about 71.2 per cent on average. From Table 7 given only dimensions of basil leaves in terms of ratio of height to width of each leaf, the obtained accuracy measurement is about 76.3 per cent on average. But from Table 8 given all characteristic features retrieved from basil leaves, the obtain accuracy measurement is about 99.7 per cent on average which is very close to 100 per cent of accuracy.

8 Discussion and Conclusion

From our experiments, the lowest accuracy which is found for only one case of experiments is obtained from the fine Gaussian SVM model. Since the fine Gaussian SVM model uses the least kernel scale, accuracy measurement is generally lower than the other models of SVM classifier. To improve performance of this fine Gaussian SVM model so that it can categorize these two basil-species better, we recommend that finer or deeper details representing the distinguishing characteristics of two species are required.

As mentioned earlier that one major advantage of the SVM classifiers is that an SVM classifier can still provide a satisfying result, especially in terms of accuracy measurement, even though it was trained using only some small training dataset as used in our case. Furthermore, there are various features on a basil leaf, of either holy basil or sweet basil, which can be utilized to characterize species of the leaf. Those features we observed including scribbled patterns of the basil leaf's edge, color shades on the basil leaf, and the leaf's dimensions.

To demonstrate our feature selection of a basil leaf and how good approaches we used to extract information of each feature from a basil leaf, we measured using the accuracy obtained from six SVM classifiers composing of linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian SVM, and coarse SVM classifier. And all six mentioned SVM classifiers showed that we achieved very high accuracy of classification, which is 100 per cent mostly, when all features were applied to generate the classifier models.

Thus, from the measured results, our proposed methods for digital information retrieval from the two basil-species sharing a common genus have been proved that the techniques we applied in feature extraction and also the basil's dominant features we selected provide almost perfect success in categorizing the two basil-species. Hence, we may call the dominant features we selected as characterized features of the two basil-species.

Acknowledgments

This work is supported by Thailand Research Fund under grant number RTA6080013.

References

[1] T. Vijayashree, A. Gopal, Comparison Procedure for the Authentication of Basil (Ocimum Tenuiflorum) Leaf Using Image Processing Technique, *Proceeding of International Conference on Communications and Signal (ICCSP)*, India, 2015, pp. 0075-0078.

[2] A. Singh, M. L. Singh, Automated Color Prediction of Paddy Crop Leaf Using Image Processing, *IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR)*, Amritsar, 2015, pp. 24-32.

[3] A. Yadav, M. K. Dutta, C. M. Travieso, J. B. Alonso, Automatic Identification of Botanical Samples of Leaves Using Computer Vision, *International Conference and Workshop on Bioinspired Intelligence (IWOBI)*, Noida, 2017, pp. 1-6.

[4] V. Bylaiah, *Leaf Recognition and Matching with MATLAB*, Ph. D. Thesis, San Diego State University, San Diego, CA, 2014.

[5] W. Yin, C. Xiang, L. Tang, S. Chen, Venation Extraction of Leaf Image by Bi-dimensional Empirical Mode Decomposition and Morphology, *IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Mumbai, India, 2015, pp. 952-956.

[6] N. D. Keni, R. Ahmed, Neural Networks Based Leaf Identification Using Shape and Structural Decomposition, *International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, Mumbai, India, 2016, pp. 225-229.

[7] P. Chaudhary, A. K. Chaudhari, A. N. Cheeran, S. Godara, Color Transform Based Approach for Disease Spot Detection on Plant Leaf, *International Journal of Computer Science and Telecommunications*, Vol. 3, No. 6, pp. 65-70, June, 2012.

[8] S. Hati, G. Sajeevan, Plant Recognition from Leaf Image Through Artificial Neural Network, *International Journal of Computer Applications*, Vol. 62, No. 17, pp. 15-18, January, 2013.

[9] K. Singh, I. Gupta, S. Gupta, SVM-BDT PNN and Fourier Moment Technique for Classification of Leaf Shape, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 3, No. 4, pp. 67-78, December, 2010.

[10] B. P. Toth, M. Osvath, D. Papp, G. Szucs, Deep Learning and SVM Classification for Plant Recognition in Content-Based Large Scale Image Retrieval, *Proceedings of Conference and Labs of the Evaluation Forum (CLEF)*, Hungary, 2016, pp. 1-10.

[11] C. Zhang, D. Wu, R. W. Liu, N. Xiong, Non-Local Regularized Variational Model for Image Deblurring Under Mixed Gaussian-Impulse Noise, *Journal of Internet Technology*, Vol. 16, No. 7, pp. 1301-1319, December, 2015.

[12] F. Ahmed, S. Das, Removal of High-Density Salt-and-Pepper Noise in Images With an Iterative Adaptive Fuzzy Filter Using Alpha-Trimmed Mean, *IEEE Transactions on Fuzzy Systems*, Vol. 22, No. 5, pp. 1352-1358, October, 2014.

[13] V. Jaouen, P. Gonzalez, S. Stute, D. Guilloteau, S. Chalon, I. Buvat, C. Tauber, Variational Segmentation of Vector-Valued Images With Gradient Vector Flow, *IEEE Transactions on Image Processing*, Vol. 23, No. 11, pp. 4773-4785, November, 2014.

[14] S. Lin, H. Ruimin, Z. Rui, Depth Similarity Enhanced Image Summarization Algorithm for Hole-Filling in Depth Image-Based Rendering, *China Communications*, Vol. 11, No. 11, pp. 60-68, November, 2014.

Biographies

Varin Chouvatut graduated with B.Eng. (Honours) and M.Eng. in Computer Engineering and got Ph.D. in Electrical and Computer Engineering from King Mongkut's University of Technology Thonburi since 2011. She is an assistant professor at Chiang Mai University. Her research interests include computer vision, image processing, computer graphics, and data science.

Ekkarat Boonchieng got Ph.D. in Computer Science from Illinois Institute of Technology since 2000. He is currently a Director of Center of Excellence in Community Health Informatics, Chiang Mai University. His research interests include computer graphics, image processing, computer network, data science and biomedical engineering.