International Journal of Engineering & Technology, 7 (2.15) (2018) 49-53

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Image Pre-Processing Algorithm for *Ficus deltoidea* Jack (Moraceae) Varietal Recognition: A Repeated Perpendicular Line Scanning Approach

Ahmad Fakhri Ab. Nasir[1]*, M Nordin A Rahman[2], Nashriyah Mat[3], A Rasid Mamat[4], Ahmad Shahrizan Abdul Ghani[5]

[1,5]Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600, Pekan, Pahang
[2,4]Faculy of Informatics and Computing, Universiti Sultan Zainal Abidin, Tembila Campus, 22000 Besut, Terengganu
[3]Faculty of Bioresources and Food Industry, Universiti Sultan Zainal Abidin, Tembila Campus, 22000 Besut, Terengganu
**Email: afakhri@ump.edu.my*

Abstract

Image pre-processing task is always the first crucial step in plant species recognition system which is responsible to keep precision of feature measurement process. Some of researchers have developed the image pre-processing algorithm to remove petiole section. However, the algorithm was developed using semi-automatic algorithm which is strongly believed to give an inaccurate feature measurement. In this paper, a new technique of automatic petiole section removal is proposed based on repeated perpendicular petiole length scanning concept. Four phases of petiole removal technique involved are: i) binary image enhancement, ii) boundary binary image contour tracing, iii) petiole section scanning, and iv) optimal image size retaining and cropping. The experiments are conducted using six varieties of *Ficus deltoidea* Jack (Moraceae) leaves. The experimental results indicate that the segmentation results are acceptably good since the digital leaf images have less than 1% of segmentation errors on several ground truth images.

Keywords: *Ficus deltoidea Jack; image pre-processing; image processing; leaf recognition; plant species recognition.*

1. Introduction

Ficus deltoidea is a large shrub or a small tree up to 3 meters tall with aerial roots that often begins its life as an epiphyte plant [1]. Figs or *Ficus* plants originated from Asia Minor and can be found throughout Mediterranean, Indian subcontinent, Latin America, Texas, Southern California, and until the Far East such as in the Malaysian tropical rain forest [2]. According to [3], seven varieties of *F. deltoidea* can be found in Peninsular Malaysia; namely *var. deltoidea*, *var. angustifolia*, *var. trengganuensis*, *var. bilobata*, *var. intermedia*, *var. kunstleri*, and *var. motleyana*. The decoction of the leaves is believed to improve blood circulation, stimulate aphrodisiac activity, and possess antioxidant as well as antidiabetic properties [4], [5].
Plant identification based on leaf provides great significance in plant taxonomy. These parameters help the taxonomists to distinguish between different species. The identification of *Ficus deltoidea* varieties is vital to plant collectors and amateur botanists in order to correct classification of this plant. The traditional method used to identify *F. deltoidea* plant varieties led the taxonomists to observe and examine the morphologies of herbarium or live specimens that was time consuming, less efficient, troublesome task for nonprofessional, and sometimes, extremely complex identification task. Hence, the existence of sophisticated technologies, such as digital cameras and computers, has led to an increasing interest in automating the process of plant species identification. Computational in plant species identification can be seen as a part of computer vision and pattern recognition area [6], [7]. It is usually implemented by using digital leaf image due to its simplicity to acquire digital image.
The basic components of pattern recognition system are data pre-processing, data measurement and representation, as well as data classification [8], [9]. In data pre-processing, the input data are pre-processed so that the data can be read and analyzed by the computer. In the real world, data are often far from perfect. Most pattern recognition techniques cannot tolerate some levels of imperfection in the image for analysis [10]. Therefore, a good image pre-processing algorithm must be modeled properly before the feature measurement process is computed from the images and further improves the pattern recognition system. The rest of the paper is organized as follows: Section 2 describes related work on image pre-processing algorithm for plant species recognition. Section 3 discusses the detailed of the image pre-processing algorithm for *Ficus deltoidea* varieties recognition. Experimental results of the proposed algorithm are presented in Section 4. Finally, conclusion is placed in Section 5.

2. Related work

In this domain, most researchers are interested in using a single leaf condition (data acquisition) for instance in [11], [12] because it is easier to obtain an accurate measurement of leaf features. In addition, a digital scanner is usually used as a device to acquire leaf images digitally for instance implemented in [13], [14]. By doing this, the size scale of the image and the lighting environment are acquired in a controlled environment.

2.1. Image pre-processing algorithm without petiole section

Most of single leaf image in the previous image pre-processing algorithm is usually acquired without the involvement of petiole section. This algorithm used grayscale histogram shape-based threshold method [12], [13], [14] to isolate the objects of interest (leaf) from other unnecessary objects (background). Therefore, the first process involved in this algorithm is to convert the original RGB image to grayscale image (process from Fig. 2 (a) to Fig. 2 (b)). Secondly, peaks and valleys of grayscale histogram are extracted and analysed. The bottommost (lowest) point (valley) which is between two highest peaks in histogram denotes the threshold value (process from Fig. 2 (b) to Fig. 2 (c)) as illustrated in Fig. 1. The threshold level is calculated by (specific threshold / 255), where 255 is the maximum value of grey level. Then, the output binary image is obtained by replaced all pixels in the input image with greater than the threshold level by the value of 1 and replaced all other pixels by the value of 0.

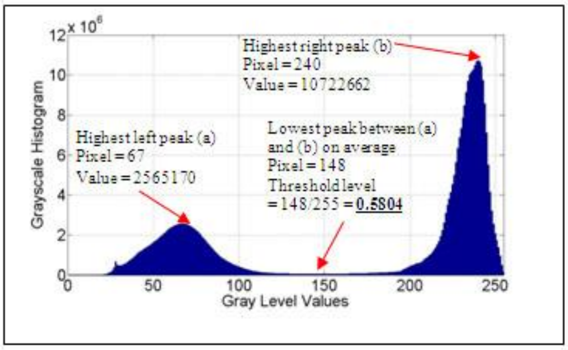


Fig. 1: Example of gray level values for all images in the database

Then, the 3x3 Laplacian filter algorithm is computed from the binary image to obtain the outer edge also as in previous section (process from Fig. 2 (c) to Fig. 2 (d)). The outer edge is usually used in the previous algorithm to calculate vein features [12], [15]. Fig. 2 represents the results example of image pre-processing algorithm without petiole section.

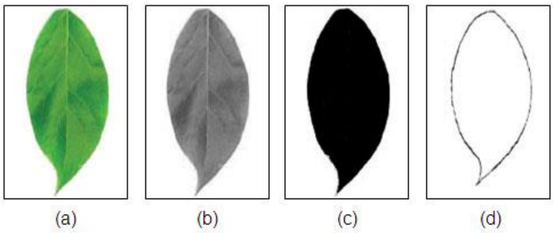


Fig. 2: Examples of image pre-processing results for algorithm without petiole section, (a) original RGB image, (b) grayscale image, (c) binary image, and (d) boundary image [12]

2.2. Image pre-processing algorithm with petiole section

Since the collected leaf images have petiole section, this section need to be removed in order to keep precision of leaf feature measurement. As referred to manual measurements of *Ficus deltoidea* varieties identification [2], the leaf length, area and width for example are calculated without the leaf petiole. To keep the precision of leaf shape features measurement, these leaf petioles should be further removed from the obtained binary images. An addition function should be inserted to the image pre-processing algorithm without petiole section to remove this petiole section. In the previous leaf petiole removal algorithms [16] and [17], the petiole removal is basically based on semi-automatic mode. However, it is believed and important to have leaf recognition system in fully-automatic mode since the semi-automatic algorithm will lead to troublesomeness for amateur botanists to complete the recognition task. For instance in [16], by using morphology closing with disk structuring element, the programmer has to select an appropriate radius of disk structuring element in order to remove petiole section. In the empirical experiment, bigger radius is needed to proper removed the petiole section and vice versa. Image closing is image erosion followed by image dilation. The mathematical definition of erosion, dilation and closing for binary images can be referred to [16].

Besides in [17], the petiole section is removed by scanning the petiole thickness from the top to the bottom of image. Rows whose thickness fell below a certain threshold (as a ratio of the maximum thickness of the leaf) were identified as petiole sections, and were removed from the image to obtain the final binary leaf image. Fig. 3 (b) shows the binary after petiole removal from Fig. 3 (a).

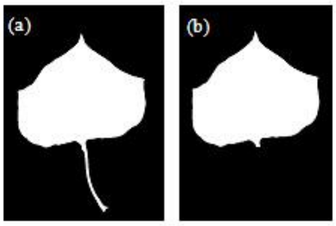


Fig. 3: Example of petiole removal results, (a) An image after the previous segmentation and (b) the binary image after petiole removal [17]

3. The image pre-processing algorithm

Fig. 4 shows the general flow of all phases in the proposed image pre-processing algorithm including the four added phases (highlighted box). The four phases of petiole removal are: Binary image enhancement, Boundary binary image contour tracing, Petiole section scanning, and Optimal image size retaining and cropping. Four output images are produced from image pre-processing algorithm and used for feature measurement: i) grayscale color of raw leaf image for calculating texture features. This image is obtained from process number 2; ii) leaf region with elimination of petiole section in binary image for computing leaf shape, and vein. The image is produced by the process number 4 to number 7; iii) leaf apex region, the image is cropped to ~25% on top of image after performing process number 7. The image is used for calculating apex angle, and; iv) leaf boundary for completing the leaf vein features calculation. The image is produced by the process number 8.

3.1. General concept of petiole section scanning

The petiole removal process was basically based on top-down perpendicular length scanning concept (Fig. 5). Top-down scanning started with calculating length of perpendicular line pl1 to perpendicular line pl_{12}. The perpendicular length of leaf contour was calculated by using the found perpendicular left and right outline of leaf object for instance in pl_1, the distance between left point a_1[116,562] and to right point a_2[116,578] is computed. Therefore, in order to get two points for each of perpendicular line, the algorithms must have an ability to trace outline of leaf object. Hence, the boundary binary image contour tracing was implemented. However, the contour tracing function is sensitive to noisy images. By comparing the detected contour points on noisy image and smoothed image, it is clearly showed that the found contour points in the noisy image were sometimes misclassified. The petiole scanning process will be wrongly calculated if the false contour points are being used. Thus, in order to reduce misclassified of the obtained contour points, binary image enhancement was added before the contour tracing process started. In this algorithm, image opening followed by closing operations using square 5×5 SE was computed on leaf images

Then, the collections of contour coordinates were obtained from contour tracing and these coordinates are sorted in the array lists from smallest y plane (horizontal plane) coordinate (top) to largest y plane coordinate (low). The detected petiole area was calculated based on specific length threshold and the sum of iterative length that appeared in contour coordinates. The calculations of this process will be elaborated in the later subsection. Finally, optimal image size retaining and cropping function was implemented for locating the base and apex sections of leaf. Normally, the base position is located 25% from the bottom of image and the apex position is located 25% from the top of image. This is very useful in calculating the base and apex angle of leaf which will be used in feature measurement phase.

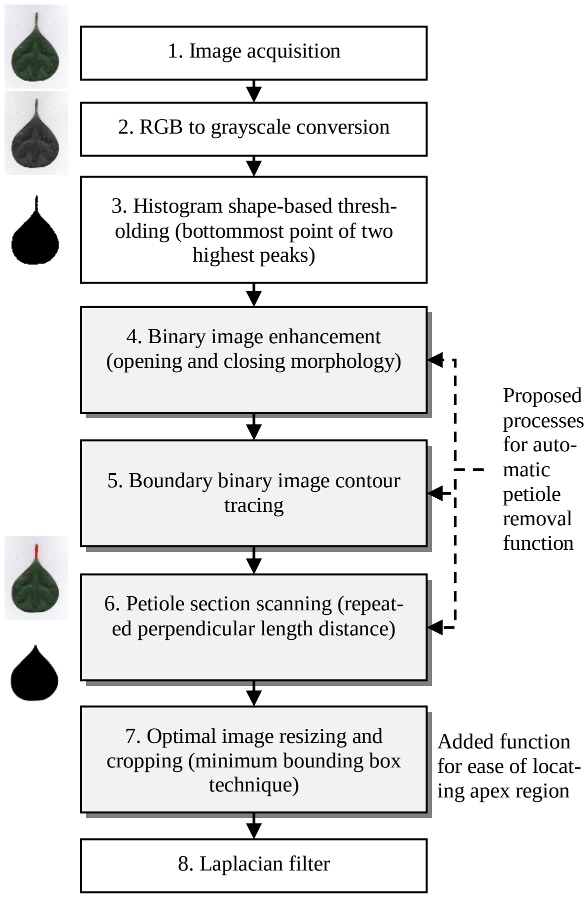


Fig. 4: The proposed image pre-processing algorithm process flow

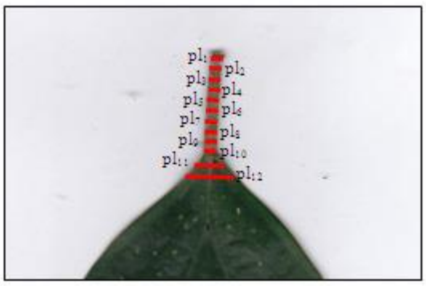
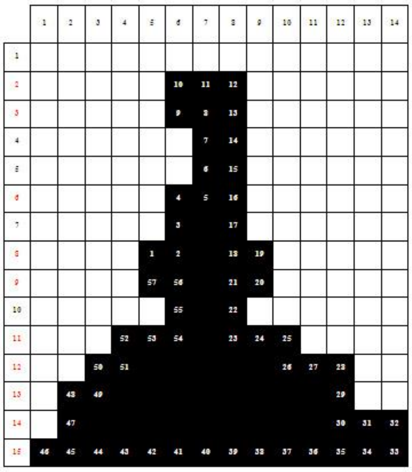


Fig. 5: The petiole removal based on top-down perpendicular length scanning

3.2. Automatic petiole section removal

The petiole section scanning was accomplished using two main steps which are as follows: i) Step 1: Calculating perpendicular length distance (leaf thickness), and; ii) Step 2: Calculating the number of repeatedly perpendicular length distance. Set the number of repeatedly perpendicular length distance.

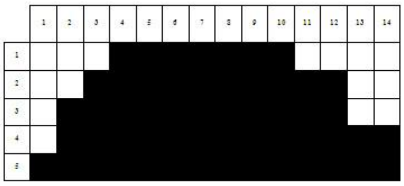
Step 1: After the boundary of leaf was obtained, the collection of boundary pixels coordinates need to be sorted to represent top to bottom of image. The sorting process was done by compiling column elements from low number to high number. By doing this process, all of boundary pixels coordinates were sorted from top to bottom. Referring to Fig. 6, on the given boundary pixels (starting point set in number 1), the sorted pixels were pixel number 10, 11, 12 (column 2), 9, 8, 13 (column 3), 7, 14 (column 4), and so on. Noticed that in some of columns, there was a number of boundary pixels that have more than two coordinates (highlighted in red font in Fig. 6). In order to find the horizontal perpendicular length, the number of boundary pixels must be only in two coordinates which are the most left and the most right. Hence, in the second process in this step, the distance between two boundary coordinates in each of column was implemented by computing only the first and the last coordinates. This distance was computed using Euclidean distance formula.



Perpendicular Distance (PD)	Repeated PD
nil	nil
2	5
2	5
1	2
1	2
2	5
2	5
4	2
4	2
2	5
6	1
9	1
10	1
12	1
13	1

Fig. 6: An illustration of petiole scanning

Step 2: Using an array containing distance of each column, the scanning process was then constructed. In this algorithm, two parameters were used: sorted repeated perpendicular length and length threshold. As example in Fig. 6, the perpendicular distance 2 was then repeated 4 times, the perpendicular distance number 1 and 4 repeated 2 times, and so forth. In the observations, the petiole section in *Ficus deltoidea* leaves was always repeated more than 7 times. Therefore, the repeated perpendicular length was computed and only the distances with 7 times repeatedly were selected as petiole sections. By considering example in Fig. 6, let's say the repeated perpendicular distance is set as ≥ 2, the obtained petiole sections is column number 2, 3, 4, 5, 6, 7, 8, 9, and 10. Other columns are considered as leaf sections.



Perpendicular Distance (PD)	Repeated PD
6	1
9	1
10	1
12	1
13	1

Fig. 7: An illustration of leaf without petiole, example from Fig. 6

In the leaf images without petiole section, for instance in Fig. 7, there was none of repeated perpendicular distance that is ≥ 2. The petiole removal process using the previous algorithm [16], [17] has to be done in semi-automatic fashion. To solve this problem, this algorithm proposed the repeated perpendicular length technique to give an ability to trace petiole section in fully-automatic mode. Fig. 8 shows the algorithm for petiole section scanning and removal.

4. Experiments and result analysis

All of image pre-processing algorithms were coded under MATLAB. The experimental objective was to obtain the segmentation performance on the developed image pre-processing algorithm. In this experiment, the proposed algorithm was not compared with other previous works because other works such as [16] and [17] are based on semi-automatic petiole removal. In addition, although algorithm [12] was built for *Ficus deltoidea* varieties recognition, however, the algorithm used digital leaf images without involvement of petiole section. Therefore, [12] algorithm also was not compared in this study.

Algorithm: Petiole section scanning
Input: Binary image (leaf and petiole regions)
Output: 1) Binary image (leaf regions)
2) Binary image (petiole regions)
1. Start
2. Step through all pixel boundaries $P_0 \ldots P_{n+1}$
3. Sort pixel boundaries based on ascending column number
4. For each of sorted image boundaries, $IB_i = (x, y)$
5. Calculate perpendicular distance, $PD_i = \sqrt{(x_{last} - x_{first})^2 + (y_{last} - y_{first})^2}$
6. For each of PD_i
7. Find repeated PD_i, rPD_i
8. If $rPD_i >= 7$ //repeated perpendicular length with more than seven (7) times
9. Binary image = Petiole regions
10. Else
11. Binary image = Leaf regions
12. Stop

Fig. 8: Pseudo-code for petiole section scanning algorithm

4.1. Experiment setup

In object detection, the term "ground truth image" always refers to the actual detected objects in the given image. The ground truth image was formulated using the Center for Digital Video Processing (CDVP) interactive segmentation tool. In the absence of standardized public ground truth database and experts in the area of research, researchers use interactive segmentation tool to construct ground truth image (GTI) for further performance evaluation of segmentation methods [18]. The CDVP tool enables users to extract objects from images, simply by marking areas of the image as "object" or "background" with the mouse. The CDVP tool was based on graph cut based segmentation method and it is a free software tool that can be operated on various operating systems.

One of the simplest though effective methods for estimating errors of segmentation is by calculating the average correct and incorrect classified pixels occurred in segmented binary image compared to GTI (real image) [19]. This method was calculated by counting the false positive (FP), false negative (FN), true positive (TP) and, true negative (TN) which is the same as supervised classification in data mining. However, in the most situations, the number of white and black pixels (predicted class) is imbalanced. Therefore, the weight or probability can be implemented for efficient errors estimation. Two segmented images (SI) were used (Fig. 9 (b) and Fig. 9 (c)) to be compared with GTI (Fig. 9 (a)). By using the naked eyes, Fig. 9 (c) should has better segmentation result because only two pixels are misclassified as background pixels (white pixels). Table 1 showed the explanations of average segmentation error (Se) calculation. The Se for SI in Fig. 9 (b) is 0.28 (or 28%), and SI in Fig. 9 (c) is 0.08 (or 8%). Thus, SI in Fig. 9 (c) showed that the implemented segmentation algorithm was more efficient as compared to SI in Fig. 9 (b). This calculation was used in the following experiments.

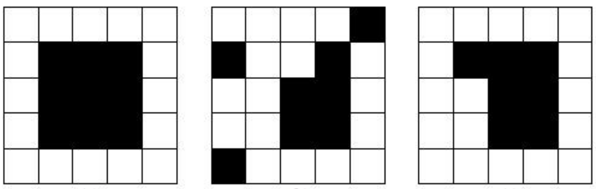


Fig. 9: (a) ground truth image, (b) segmented image no. 1, and (c) segmentation image no. 2

4.1. Result analysis

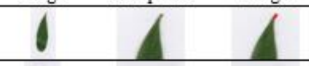
The process for creating ground truth information for real images is very time consuming since the pixels in the image is label manually. Even though CDVP tool is very helpful (in semi-automatic run), however with 420 images in leaf database, it is also time consuming efforts. Therefore, only two images of each variety were randomly picked in the leaf database and used in the experiment. In total, 12 leaf images (see Table 2) were selected for the image pre-processing algorithm experiments. Using the randomly picked leaf images, Se for leaf region is calculated using methodology as showed in Table 1. In this case, Se for petiole region was assumed similar to Se for leaf region. In other words, lower Se for leaf region contributes to lower Se for petiole region and vice versa. Therefore, for an easy construction of GTI, only leaf region was used to represent the efficiency of the algorithm. In general, the lower the Se corresponds to better segmentation method.

Table 1: Segmentation error calculations

Definition	Denoted as	Formula
Probability of each level	$P(black)$	$\frac{\text{number of black pixel in GTI}}{\text{total number of all pixels in GTI}}$
	$P(white)$	$\frac{\text{number of white pixel in GTI}}{\text{total number of all pixels in GTI}}$
False positive and negative	FP	$\frac{\text{number of black pixel in GTI but white pixel in SI}}{\text{number of black pixel in GTI}} \times P(black) \times 100$
	FN	$\frac{\text{number of white pixel in GTI but black pixel in SI}}{\text{number of white pixel in GTI}} \times P(white) \times 100$
True positive and negative	TP	$\frac{\text{number of black pixel in GTI and black pixel in SI}}{\text{number of black pixel in GTI}} \times P(black) \times 100$
	TN	$\frac{\text{number of white pixel in GTI and white pixel in SI}}{\text{number of white pixel in GTI}} \times P(white) \times 100$
Segmentation error	SE	$\frac{[FN]+[FP]}{[FN]+\{FP\}+[TN]+[TP]}$

As referred to Table 2; images 3.jpg, 32.jpg, 1.jpg, 23.jpg, 26.jpg, 57.jpg, 39.jpg, 62.jpg, 10.jpg, 25.jpg, 15.jpg, and 45.jpg have segmentation error of 0.12%, 0.22%, 0.12%, 0.00%, 0.18%, 0.43%, 0.14%, 0.21%, 0.32%, 0.50%, 0.20%, and 0.36%, respectively. On the average, segmentation error for the several selected images was 0.23%. The experimental results showed that the proposed algorithm is able to detect the image with normal straight petiole section (i.e. 26.jpg), the image with no petiole section (i.e. 23.jpg), the image with long petiole section (i.e. 25.jpg), the image with short petiole section (i.e. 1.jpg), and the image with slanting petiole section (i.e. 10.jpg), with a very minimal segmentation error (<1%).

Table 2: Segmentation errors for the image pre-processing algorithm

Variety	Image no.	Original full image	Original image zoom on top side	Detected petiole image	Se
angustifolia	3.jpg				0.12%
	32.jpg				0.22%
bilobata	1.jpg				0.12%
	23.jpg				0.00%
deltoidea	26.jpg				0.18%
	57.jpg				0.43%
intermedia	39.jpg				0.14%
	62.jpg				0.21%
kunstleri	10.jpg				0.32%
	25.jpg				0.50%
trengganuensis	15.jpg				0.20%
	45.jpg				0.36%

5. Conclusion

This paper introduces a newly developed image pre-processing algorithm which provides an automatic and efficient way to isolate petiole and leaf sections. The petiole removal function was developed by an expansion of the top-down perpendicular length scanning concept. A repeated perpendicular length scanning mechanism was proposed to scan the petiole section. Only the repeated perpendicular length with more than 7 times is considered as petiole region. Using the added four phases in petiole removal function, the algorithm has successfully segmented the petiole and leaf regions with average segmentation error of only 0.23%. The result indicates that the algorithm can be recognized as highly efficient (will accurately measure the leaf features in the next process of plant species recognition) since the misclassified leaf section was only below than 1%. The developed image pre-processing algorithm was applicable to be implemented in other plant species recognition system that needs isolation of leaf petiole in fully automatic mode and further select the petiole as feature.

Acknowledgement

The authors duly acknowledged the permission of the Faculty of Bioresourses and Food Industry, Universiti Sultan Zainal Abidin (UniSZA), Besut, Terengganu, Malaysia for allowing us to pick and capture some leaves of *Ficus deltoidea* at the faculty nursery.

References

[1] Corner EJH (1996), The complex of Ficus deltoidea; a recent invasion of the Sunda Shelf. *Philosophical Transactions of the Royal Society of London* 256, 281–317.
[2] Nashriyah M, Nurrul Akmar R, Nor Zaimah AR, Norhaslinda H, Zanariah MN, Nur Fatihah HN, Abd Ghani Y & Abdul Manaf A (2012), Leaf Morphological Variations and Heterophyllyin Ficus deltoidea Jack (Moraceae). *Sains Malaysiana* 41, 5, 527–538.
[3] Turner IM (1995), Catalogue of the vascular plants in Malaya. *Garden's Bulletin Singapore* 47, 2, 347–757.
[4] Norhaniza A, Sin CY, Chee ES, Nee KI & Renxin L (2007), Blood Glucose Lowering Effect of Ficus deltoidea Aqueous Extract. *Malaysian Journal of Science* 26, 1, 73–78.
[5] Sulaiman MR, Hussain MK, Zakaria ZA, Somchit MN, Moin S, Mohamad AS & Israf DA (2008), Evaluation of the antinociceptive activity of Ficus deltoidea aqueous extract. *Fitoterapia* 79, 557–561.
[6] Kumar N, Belhumeur PN, Biswas A, Jacobs DW, Kress WJ, Lopez I & Soares JVB, Leafsnape: A Computer Vision System for Automatic Plant Species Identification. 1*2th European Conference on Computer Vision*, 2012, pp:502–516.
[7] Satti V, Satya A & Sharma S (2013), An automatic leaf recognition system for plant identification using machine vision technology. *International Journal of Engineering Science and Technology* 5, 4, 874–879.
[8] Jain AK, Duin PW & Mao J (2000), Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligent* 22, 1, 4–37.
[9] Polikar R, *Pattern Recognition*, John Wiley and Sons, New York, (2006), pp:1–22.
[10] Chakrabarti S, *Data Mining Know It All*, Morgan Kaufmann Publishers, San Francisco, CA, (2009).
[11] Kadir A, Nugroho LE, Susanto A & Santosa PI (2012), Performance Improvement of Leaf Identification System Using Principal Component Analysis. *International Journal of Advanced Science and Technology* 44, 113–124.
[12] Asnor JI, Azura CS, Mohammad Hamiruce M, Shamsul K & Mohammad Ali JG, Automated Recognition of Ficus Deltoidea Using Ant Colony Optimization Technique. *8th IEEE Conference on Industrial Electronics and Applications*, 2013, pp:296–300.
[13] Lee CL & Chen SY (2006), Classification of Leaf Images. *International Journal of Imaging Systems and Technology* 16, 1, 15–23.
[14] Du JX, Wang XF & Zhang GJ (2007), Leaf shape based species recognition. *Applied Mathematics and Computation* 185, 883–893.
[15] Wu SG, Bao FS, Xu EU, Wang YX, Chang YF & Xiang QL, A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. *International Symposium on Signal Processing and Information Technology*, 2007, pp:11–16.
[16] Wang XF, Huang DS, Du JX, Xu H & Heutte L (2008), Classification of plant leaf images with complicated background. *Applied Mathematics and Computation* 205, 916–926.
[17] Arora A, Gupta A, Bagmar N, Mishra S, Bhattacharya A, A Plant Identification System using Shape and Morphological Features on Segmented Leaflets: Team ITK, CLEF 2012. *Online working notes, Evaluation Labs and Workshops*, 2012.
[18] McGuinness K & O'Connor NE (2010), A Comparative Evaluation of Interactive Segmentation Algorithms. *Pattern Recognition* 43, 2, 434–444.
[19] Estrada FJ & Jepson AD, Quantitative Evaluation of a Novel Image Segmentation Algorithm. *Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp:132–113.