Improving Leaf Classification Rate via Background Removal and ROI Extraction

Yen-Ju Wu and Chun-Ming Tsai
Department of Computer Science, University of Taipei, Taiwan, R.O.C.
Email: {shiyoutai.w, cmtsai2009}@gmail.com

Frank Shih
Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102-1982, USA
Email: shih@njit.edu

***Abstract*—Modern description methods are used for plant classification through leaf recognition. These methods usually include color transformation, feature detection and description, dimension reduction, and classification. However, these methods use an original image as the input image from which to extract the features to be recognized. In this condition, computational complexity will increase. To reduce computational time, in the proposed method the Region of Interest (ROI) is extracted before extracting features from the image. Quality of image also plays an important role in increasing leaf classification rate. A good quality image gives better classification rate than noisy images. To extract features exactly from noisy images is very difficult which in-turn reduces leaf classification rate. To overcome problems occurring due to noisy image quality, background removal is done before extracting features from the image. That is, the proposed method includes color transformation, preprocessing (background removal and ROI extraction), feature extraction and description, and classification. In experiments and comparing results with and without preprocessing methods, the proposed method gives classification rate with an accuracy greater than 92.13% and the computational time in average is 133.94ms per leaf image.**

***Index Terms*—leaf classification, background removal, ROI extraction, feature extraction, codebook creation**

I. Introduction

Plants play a very important role in the ecological environment: they can absorb light energy and carbon dioxide in photosynthesis to produce oxygen and provide aerobic biological survival on Earth. For humans, there are a variety of benefits from plants, such as: food, medical, construction, fuel and many others. In addition, plants can sustain the climate balance and stability of various ecosystems. Currently, there are about 400,000 species of plants in the world and about 350,000 kinds are flowering plants [1]. However, growth of population and the environmental pollution caused by industrialization have gradually harmed plant species survival. Therefore, it will become more and more important to identify plant species.

Manuscript received September 20, 2015; revised July 10, 2016.

Most of the plant species identification methods extract features from whole leaf. However, in this condition, there are two shortcomings: (1) the same class leaves have different features, which will affect the classification accuracy. (2) The computational time will increase when extracting feature from the whole leaf. In this paper, to address these shortcomings, background removal is used to extract the Region of Interest (ROI) from the original image. Next, the keypoint-based features are extracted from the ROI image. Third, bag of words is coded from these keypoint features. Last, the support vector machine (SVM) is used to classify the leaf. Fig. 1 shows the flow diagram of the proposed method.

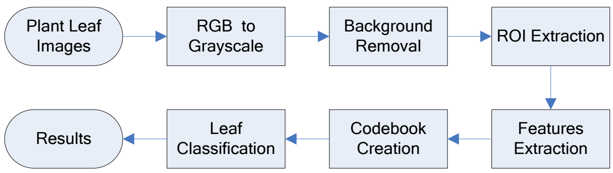


Figure 1. Flow diagram of the proposed method.

The rest of the paper is organized in the following fashion: the related works are discussed in Section 2. The image preprocessing is presented in Section 3. The feature extraction and codebook creation are described in Section 4. The experimental results are illustrated in Section 5. The conclusion and the future work are exposed in Section 6.

II. Related Works

Cope *et al.* [2] reviewed plant species identification using digital morphometrics. From their reviews, most of the methods and applications used stems, leaves, buds, and fruit to identify plant species. However, some species may not have flower or fruit, so many studies are based on the leaves to identify plant species. Furthermore, many methods used shape, color, and texture to identify the plant leaves.

Traditional methods to recognize plant species are carried out by manual matching of the plant's features, which include leaf, flower, stem, root, and fruit. Leaves of each plant species carry useful information for classification of various plants. Because leaves are

doi: 10.18178/joig.4.2.93-98

relatively constant over the season, leaves are the most suitable characteristic to distinguish among plant, and shape of leaf is the most important particular characteristic for plant species identification. Leaves from a species differ in size, shape, and color, but their typical characteristic is constant. Furthermore, leaves are easy to find, collect and keep for digital images. The data collection process (i.e., collecting leaf samples) is quite simple.

A specimen leaf can easily be converted to a leaf image, and features can be extracted automatically to recognize species by using image processing techniques [2].

Wu *et al.* [3] combined five basic features to obtain 12 digital morphological features, principal components analysis, and probabilistic neural network methods to implement a leaf recognition system for plant classification. In their experiments, their system was trained by 1800 leaves to classify 32 kinds of plants with accuracy rate greater than 90%.

Bama *et al.* [4] used shape, color and texture features to propose a plant image retrieval method based on plant leaf images. The SIFT method is used to extract shape-based, color-based, and texture-based features. They used precision and recall to test their system on their collected leaf images.

Kadir *et al.* [5] noted that foliage plants have various colors and unique patterns in the leaf, and used polar Fourier transform, color moments, and vein features, to identify plant species. Their system had an accuracy rate of 93.13% when it was tested using the Flavia leaf dataset [3].

Wang *et al.* [6] took both global features and local features to improve leaf image classification. They used shape context as the global feature and used SIFT descriptors as the local feature. Finally, the weight K-NN algorithm was used to classify the leaf images. Their system had accuracy rate of 91.30% when it was tested using ICL leaf dataset. However, they have not shared their leaf database to let other researchers to use it.

Mouine *et al.* [7] proposed a new multiscale shape-based approach for leaf image retrieval. They studied four multiscale triangle representations: the Triangle Area Representation (TAR), the Triangle Side Lengths representation (TSL), Triangle Oriented Angles (TOA), and Triangle Side Lengths and Angle representation (TSLA). From their recall/precision curves, it was shown that the angular information (TSLA, TOA) enhances the retrieval performance when used with the Flavia leaf dataset [3].

Kulkarni *et al.* [8] used shape, vein, color, texture, and Zernike moments features to recognize and identify plants. They used a dual stage training algorithm to train a Radial Basis Probabilistic Neural Network (RBPNN) classifier. Their simulation results on the Flavia leaf dataset [3] show that the proposed method for leaf recognition yields an accuracy rate of 93.82%.

Mahdikhanlou and Ebrahimnezhad [9] used a centroid distance and axis of least inertia method for plant leaf classification. The Probabilistic Neural Network (PNN) has been used as a classifier in their method. Their leaf classification yields an accuracy rate of 81.5%.

Kazerouni *et al.* [10] proposed a procedure to recognize and identify plants through leaves by using bag of words (BoW) and Support Vector Machine (SVM). Their proposed method uses a Scale Invariant Feature Transform (SIFT) method and two combined methods: HARRIS-SIFT and FAST-SIFT. Their experimental results on the Flavia leaf dataset [3] shows that the accuracy rate of SIFT method is higher than other methods which is 89.3519 %. Also, the computational performance of SIFT, FAST-SIFT, HARRIS-SIFT methods are 780.43ms, 610.39ms, and 771.87ms, respectively. However, in their preprocessing stage, they only convert the color image into gray image. This reduced preprocessing will cause the correction of the feature extraction and description.

Oluleye *et al.* [11] combined a GA-based CNN edge detector and a RBF learning system for automatic classification of plant leaves. Their experimental results on the Flavia leaf dataset [3] show that their proposed method is more efficient than Canny, LoG, Prewitt, and Sobel edge detector in terms of speed and classification accuracy. Their computational time is 7.77 seconds and their classification accuracy is 90.45%.

Kalyoncu and Toygar [12] used segmentation, a combination of new and well-known feature extraction and classification methods to classify plant leaves. In particular, they used geometric features, Multi-scale Distance Matrix (MDM), moment invariants, convexity, perimeter ratio, average margin distance, and margin statistic features to distinguish leaf margins and used Linear Discriminant Classifier for classification. Their system includes preprocessing (segmentation, noise reduction, contour extraction, and corner region detection), features extraction from the binary images and contour data tracing, and classification. In their experimental results, their system has better performance and has computational efficiency. However, running time of their proposed method in seconds over Flavia dataset [3] is 356 seconds.

III. Image Preprocessing

A. *Converting RGB Image to Grayscale Image*

In order to reduce the computational time, all the input leaf images are transformed from RGB images to grayscale images. The transformation equation is showed as follows:

$$Grayscale = 0.2989R + 0.5780G + 0.1140B \quad (1)$$

where R, G, and B are the red, green, and blue components of the input RGB image, respectively.

B. *Removing Background by Using GrabCut*

In order to reduce the computational time and to extract the region of interest (RoI), GrabCut is used to remove the background of the leaf images. GrabCut is proposed by Rother *et al.* [13]. GrabCut is extended from graph-cut approach [14] with (1) optimization iterative, (2) simplification of the user interaction needed, and (3) border matting. The procedures of the GrabCut method include: (1) user draws a rectangle around the foreground region. (2) GrabCut algorithm segments this rectangle region iteratively to get the best result. (3) If the segmentation result is not fine enough, some foreground region must be marked as background for better results in the next iteration. GrabCut segmentation method is based on Gibbs energy, as in

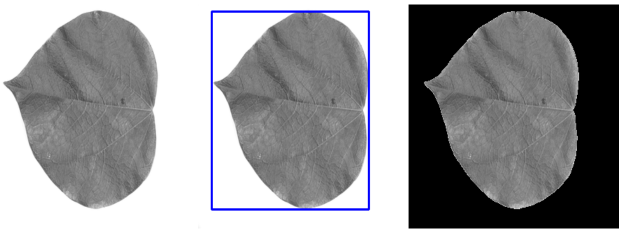
$$G(\alpha,k,\theta,z) = U(\alpha,\kappa,\theta,z) + V(\alpha,z) \quad (2)$$

where α, $\mathbf{k}$, θ, $\mathbf{z}$ represents the background (0) and the foreground (1), the GMM with k components, the parameters of the GMM, and the image pixels, respectively. The data term U is taken from the color GMM models, as in

$$U(\alpha,\kappa,\theta,z) = \sum_n D(\alpha_n, k_n, \theta, z_n) \quad (3)$$

where U represents the penalty when a pixel is classify as foreground or background. The smoothness term V is the discontinuous penalty between two adjacent pixels. If the difference between two adjacent pixels is small, the V term is small because the two pixels belong to the same foreground or background. If the difference is large, the V term is large because the two pixels are located at the boundary between the foreground and background. GrabCut uses iterative minimization and min cut for minimization of the total energy E.

Fig. 2 shows an example of the background removing by using GrabCut. The original leaf image, rectangle image, and background removal image are show in Fig. 2(a), 2(b), and 2(c), respectively. After having applied GrabCut method, the background image is shown by black color, as in Fig. 2(c).



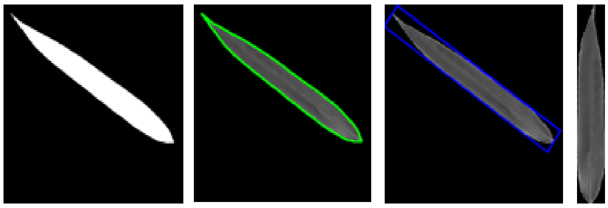
(a) (b) (c)

Figure 2. Example of background removal, (a) grayscale image, (b) A rectangle containing foreground object is set, (c) removal background image.

C. Extracting Region of Interest and Perspective Transformation

After using the GrabCut method, the background removal image is still the gray image. In order to reduce the feature extraction time and extract the Region of Interest (RoI) of the leaf, Otsu thresholding is applied to threshold the background removal image into a binary image. After above procedures, a contour finding method is applied to find the contour of the binary image of the leaf RoI. And then based on these contours, the bounding rectangle of the leaf RoI will be extracted. Because the leaves in the images have different appearance models, for example, they may be skewed or not. In order to enhance the classification accuracy rate, the perspective transformation is applied to transform the skewed image to the normal appear model.

Fig. 3 shows an example of the RoI extraction and perspective transformation. The binary image, the contour image, the bounding rectangle image, and the RoI extraction image are shown in Fig. 3(a), 3(b), 3(c), and 3(d), respectively.



(a) (b) (c) (d)

Figure 3. Example of RoI extraction and perspective transformation, (a) binary image, (b) contour image, (c) bounding rectangle image, (d) RoI extraction image.

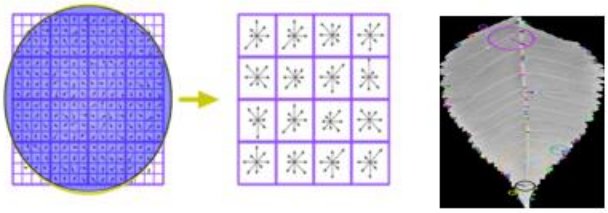
IV. Feature Extraction and Codebook Creation

A. Low-Level Feature Extraction

The leaf images have different sizes, different rotations, different illuminations, and different perspective invariances. To detect the leaf with above-mentioned characteristics, the scale-invariant key-points (SIFT) [15] feature is applied to detect the feature of the leaf for classification.

SIFT feature used to classify objects has been proved very effective. It is invariant to affine transformation. The procedure of extraction using SIFT feature is as follows. (1) Detection of scale-space extrema: the extrema are detected by different scale and Gaussian filters to convolve the leaf image, the difference of successive Gaussian-blurred images are computed. The extrema candidates are detected at the maximum/minimum of the Differences of Gaussians (DoG). (2) Accurate keypoint localization: the Taylor series expansion of scale space is used to get more accurate location of the extrema. The contrast and edge thresholds are used to remove the low-contrast and edge keypoints, respectively. (3) Orientation assignment: a region around the keypoint location is taken and the gradient magnitude and direction in this region is computed. The orientation histogram with 36bins of 360 degrees is taken. The highest peak of the histogram is chosen as the keypoint. (4) Keypoint descriptor: A 16x16 region around the keypoint is divided into 16 sub-regions of 4x4 size. For each sub-region, an 8 bin orientation histogram is obtained and a total of 8x4x4 bin values are used to form each keypoint descriptor.

Fig. 4 shows an example of SIFT keypoints extraction. The image gradient, keypoint descriptor, and SIFT descriptor in a leaf image are shown in Fig. 4(a), 4(b), and 4(c), respectively.

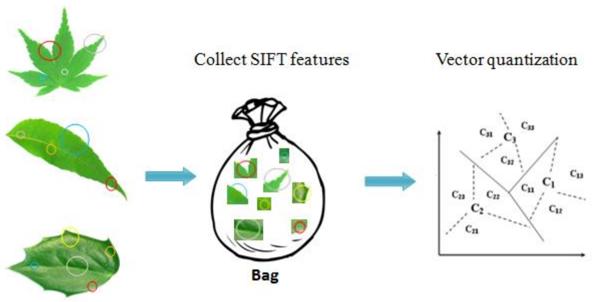


(a) (b) (c)

Figure 4. Example of SIFT keyponts extraction, (a) image gradients [15], (b) keypoint descriptor [15], (c) SIFT descriptor in a leaf image.

B. *Codebook Creation by Bag of Words*

In order to reduce the computation time for image classification, a codebook of the RoI leaf image is created. This codebook is created by the bag of keypoints. The bag of keypoints [16] is similar to the Bag of Words (BoW) representation for text categorization. BoW mode classifies based on a histogram of the frequency of visual words. In this context, the BoW is based on a histogram of the frequency of visual leaves. To use the BoW in here, the k-means clustering is used to classify the SIFT keypoint descriptor into M clusters analogous to build a codebook with M vocabularies. The flow diagram of building BoW is shown in Fig. 5. The SIFT keypoints, bag of features, and vector quantization by k-mean clustering are shown in Fig. 5(a), 5(b), and 5(c), respectively. After above-mentioned method, the leaf image is represented by the histogram of the visual words.



(a) (b) (c)

Figure 5. The flow diagram of building BoW, (a) SIFT keypoints, (b) bag of features, (c) vector quantization by k-mean clustering.

V. Experimental Results

This study system was implemented in Python 2.7, OpenCV 2.4.10 and 3.0 on an Intel(R) Core(TM) i7-3770U CPU @ 3.4GHz with Windows 7 and 8GB memory.

In order to demonstrate the proposed method, the Flavia dataset [3] is used as the experimental data. In the Flavia dataset, there are 32 different leaves of the plant images and has total 1907 leaf images. Some examples are shown in Fig. 6.

The Flavia dataset in this experiment is divided into training set with 83.3% and testing set with 16.7%. The support vector machine [17] with Radial Basis kernel is used. After optimization, the C parameter for this kernel is 8 and the γ parameter is dependent on the size of the codebook.

Figure 6. Some leaf images per species from Flavia dataset.

In order to obtain the optimal classification accuracy rate, the codebook size is varied from 158 to 758 and step 100. Using SVM to train these codebook sizes, the resulting γ parameter of the SVM for codebook sizes 158, 258, 358, 458, 558, 658, and 758 are 0.21025, 0.13125, 0.11205, 0.07001, 0.07015, 0.10015, and 0.07052, respectively. These tested results are shown in Fig. 7. From this figure, it can be seen that when the codebook size is 358, the optimal classification accuracy rate is 91.03%. After this highest point, when codebook size increases, classification accuracy rate is decreased.

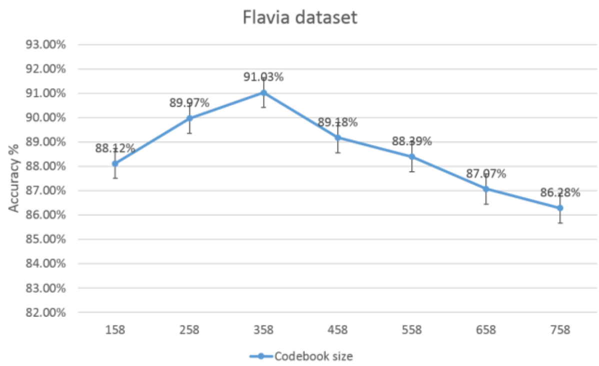


Figure 7. Curve of the classification accuracy rate with different codebook sizes.

To verify the effectiveness of the proposed method, many methods that used the Flavia dataset [3] are applied to compare the classification accuracy rates obtained. Table I shows the comparison results from different state-of-the-art methods with their classification accuracy rates.

TABLE I. Accuracy Comparison

Methods	Training samples	Testing samples	Accuracy rates
Wu *et al.* [3]	83.3%	16.7%	90.312%
Kadir *et al.* [5]	40	10	93.13%
Kulkarni *et al.* [8]	40	10	93.82%
Kazerouni *et al.* [10]	66%	34%	89.3519 %
Oluleye *et al.* [11]	83.3%	16.7%	90.45%
Kalyoncu and Toygar [12]	83.3%	16.7%	About 95%
Proposed	83.3%	16.7%	91.03%
Proposed	66%	34%	92.13%

From Table I, there are three cases for training and testing samples in experiments: (1) training samples 83.3% and testing samples 16.7%, with 10 samples per class, (2) training samples are 40 and testing samples are 10, and (3) training samples 66% and testing samples 34%. When compared to case (1), the accuracy rate of the proposed system is less than the method in [12]. However, the accuracy rate of proposed method is greater than the methods in [3] and [11]. When compared to case (2), the accuracy rate of proposed system is less than the methods in [5] and [8]. When compared to case (3), the accuracy rate of the proposed system is greater than the method in [10].

Regarding the computational time, the method in [10] is used to compare with the proposed system and as shown in Fig. 8. The computational time of testing per leaf image by the proposed system is 133.94ms. From the comparison in Fig. 8, the computational time of the proposed system is smaller than in the methods in [10].

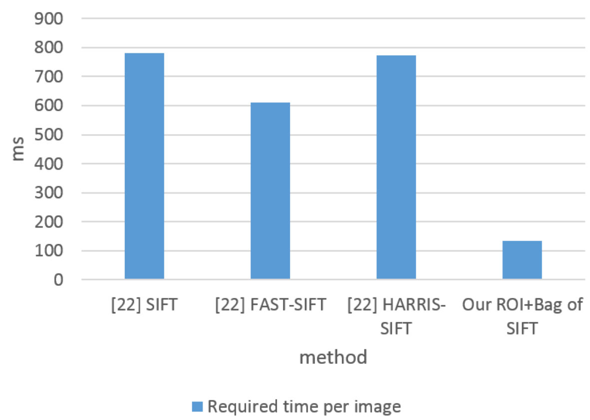


Figure 8. The computational time of testing per image.

The proposed method is similar to the method in [10]. However, the main differences between the proposed method and the method in [10] are as follows:

1) The image preprocessing of the method in [10] is only converting the RGB image into the grayscale image. The image preprocessing of the proposed method includes the RGB image into the grayscale image, background removal, and RoI extraction.
2) The feature of the method in [10] is extracted from the whole input grayscale image. The feature of the proposed method is extracted from the RoI grayscale image.
3) The codebook size of the method in [10] is fixed 1000. The codebook size of the proposed method is varying from 158 to 758 and step 100.
4) The method in [10] used υ-Support Vector Classification and the proposed method used LIBSVM [17].
5) The methods in [10] used i7-4790K, CPU @ 4.00GHz and RAM 16.0GB and the proposed method used i7-3770U CPU @ 3.4GHz and RAM 8GB.
6) The classification accuracy of the proposed method is greater than the method in [10].
7) The computational time of the proposed method is less than the method in [10].

VI. Conclusion

In this paper, improving leaf classification rate with image preprocessing is proposed. The input color leaf image is converted into grayscale image. The region of interest (leaf) in the input grayscale image is detected, cropped, normalized, and extracted. Next, the BoW of the RoI (leaf) is extracted by detecting SIFT keypoints, extracting the SIFT descriptor, and clustering codebook. Finally, the proposed classification model is trained by SVM on the BoW of the RoI dataset and the classification model is then used to test on the testing dataset. The proposed method was tested on a Flavia dataset and was compared with state-of-art methods. From the experiment results demonstrates that the proposed method has the better performance in classification accuracy rate and computational times.

In the future, in order to enhance the leaf accuracy rate, more effective features will be used and more effective classification will be used.

Acknowledgment

The authors would like to express their gratitude to Dr. Jeffrey Lee and Walter Slo-combe, who assisted editing the English language for this article. This work was supported by the Ministry of Science and Technology, R.O.C., under Grants MOST 104-2221-E-845-003-.

References

[1] Botanic Gardens Conservation International. Plant species numbers. [Online]. Available: http://www.bgci.org/ourwork/1521

[2] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: A review," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7562-7573, Jun. 2012.

[3] S. G. Wu, F. S. Bao, E. Y. Xu, Y. X. Wang, Y. F. Chang, and Q. L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *Proc. IEEE International Symposium on Signal Processing and Information Technology*, Dec. 2007, pp. 11-16.

[4] B. S. Bama, S. M. Valli, S. Raju, and V. A. Kumar, "Content based leaf image retrieval using shape, color and texture features," *Indian Journal of Computer Science and Engineering*, vol. 2, no. 2, pp. 202-211, Apr. 2011.

[5] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, "Foliage plant retrieval using polar Fourier transform, color moments and vein features," *Signal & Image Processing: An International Journal*, vol. 2, no. 3, Sep. 2011.

[6] Z. Y. Wang, B. Lu, Z. Chi, and D. G. Feng, "Leaf image classification with shape context and SIFT descriptors," in *Proc. International Conference on Digital Image Computing: Techniques and Applications*, Dec. 2011, pp. 650-654.

[7] S. Mouine, I. Yahiaoui, and A. Verroust-Blondet, "A shape-based approach for leaf classification using multiscale triangular representation," in *Proc. 3rd ACM International Conference on Multimedia Retrieval*, Apr. 2013, pp. 127-135.

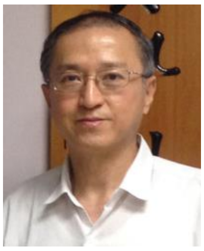
[8] A. H. Kulkarni, H. M. Rai, K. A. Jahagirdar, and P. S. Upparamani, "A leaf recognition technique for plant classification using RBPNN and Zernike moments," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 984-988, Jan. 2013.

[9] K. Mahdikhanlou and H. Ebrahimnezhad, "Plant leaf classification using centroid distance and axis of least inertia method," in *Proc.*

22nd Iranian Conference on Electrical Engineering, May 2014, pp. 1690-1694.
[10] M. F. Kazerouni, J. Schlemper, and K. D. Kuhnert, "Comparison of modern description methods for the recognition of 32 plant species," *Signal & Image Processing: An International Journal*, vol. 6, no. 2, pp. 1690-1694, Apr. 2015.
[11] B. Oluleye, A. Leisa, D. Dean, and J. S. Leng, "A neuronal classification system for plant leaves using genetic image segmentation," *British Journal of Mathematics & Computer Science*, vol. 9, no. 3, pp. 261-278, May 2015.
[12] C. Kalyoncu and O. Toygar, "Geometric leaf classification," *Computer Vision and Image Understanding*, vol. 133, pp. 102-109, 2015.
[13] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut – Interactive foreground extraction using iterated graph cut," *ACM Trans. on Graphics - Proc. of ACM SIGGRAPH*, vol. 23, no. 3, pp. 309-314, August 2004.
[14] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *Proc. International Conference on Computer Vision*, July 2001, pp. 105–112.
[15] D. Lowe, "Distinctive image features from scale-invariant key-points," *Int'l J. Computer Vision*, vol. 2, no. 60, pp. 91-110, 2004.
[16] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. "Visual categorization with bags of keypoints," in *Proc. Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004.
[17] C. C. Chang and C. J. Lin. (2003). LIBSVM—A library for support vector machines. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html

Yen-Ju Wu received his B.S. in Computer Science and Information Engineering from the Chien Hsin University, Zhongli, Taiwan, in 2013, and presently he is doing master degree at the department of Computer Science, University of Taipei.
His area of research interest is Computer Vision and Machine Learning.

Chun-Ming Tsai received the B. S. degree in Industry Education (with major in electrical engineering) from National Taiwan Normal University, Taipei, Taiwan in 1988, the M. S. and Ph. D degrees in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan in 1996 and 2002, respectively. He was a visiting scholar in Department of Computer Science, University of California Santa Barbara, California, USA, 2005, 8~2006, 7 and in Robotics Laboratory, Department of System Cybernetics, Hiroshima University, Japan, 2013, 2 ~ 2013, 4.
Dr. Tsai joined the faculty of National Tam-Shui Vocational High School, Taipei, Taiwan in August 1987. August 2003, Dr. Tsai joined the faculty of Taipei Municipal Teacher College (TMTC), Taipei, Taiwan. He was an assistant professor in the Department of Mathematics and Computer Science Education, Taipei Municipal University of Education (TMUE) from August 2006 to July 2007. He was an assistant professor in the Department of Computer Science, Taipei Municipal University of Education (TMUE) from August 2007 to July 2008. He was an associate professor in the Department of Computer Science, Taipei Municipal University of Education (TMUE) from August 2008 to January 2012. He was a professor in the Department of Computer Science, Taipei Municipal University of Education (TMUE) from February 2012 to July 2013. He is currently a professor in the Department of Computer Science, University of Taipei (UT) from August 2013. His current research interests include Video and Image Processing (VIP), Computer Vision (CV), Machine Learning (ML), Medical Image Processing (MIP), and Document Image Analysis (DIA).



Frank Shih received B.S. from National Cheng Kung University, Tainan, Taiwan, in 1980, M.S. from State University of New York, Stony Brook, U.S.A., in 1983, and Ph.D. from Purdue University, West Lafayette, IN, U.S.A., in 1987. He is presently a professor in the Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey. He is also the Director of Computer Vision Laboratory. Dr. Shih held a visiting professor position at Princeton University, Columbia University, National Taiwan University, National Institute of Informatics, Tokyo, Conservatoire National Des Arts Et Metiers, Paris, and Nanjing University of Information Science and Technology, China. He is an internationally renowned scholar and serves as the Editor-in-Chief for the International Journal of Multimedia Intelligence and Security (IJMIS). Dr. Shih is currently on the editorial board of Pattern Recognition, the International Journal of Pattern Recognition and Artificial Intelligence, Journal of Information Hiding and Multimedia Signal Processing, Recent Patents on Engineering, Recent Patents on Computer Science, The Open Nanoscience Journal, the International Journal of Internet Protocol Technology, the Journal of Internet Technology, ISRN Signal Processing, and ISRN Machine Vision. He has served as a steering member, committee member, and session chair for numerous professional conferences and workshops. He has received numerous grants from National Science Foundation, Navy and Air Force, and Industry. He was the recipient of Research Initiation Award of NSF and Board of Overseers Excellence in Research Award of NJIT.
Dr. Shih has made significant contributions to information hiding, focusing on the security and robustness of digital watermarking and steganography. He authored three books: "Digital Watermarking and Steganography," "Image Processing and Mathematical Morphology," and "Image Processing and Pattern Recognition," and edited a book: "Multimedia Security: Watermarking, Steganography, and Forensics." He published 122 journal papers, 97 conference papers, and 22 book chapters. His current research interests include image processing, computer vision, watermarking and steganography, digital forensics, sensor networks, pattern recognition, bioinformatics, information security, robotics, fuzzy logic, and neural networks.

