A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network

Stephen Gang Wu[1], Forrest Sheng Bao[2], Eric You Xu[3], Yu-Xuan Wang[4], Yi-Fan Chang[5] and Qiao-Liang Xiang[4]
[1] Institute of Applied Chemistry, Chinese Academy of Science, P. R. China
[2] Dept. of Computer Science, Texas Tech University, USA
[3] Dept. of Computer Science & Engineering, Washington University in St. Louis, USA
[4] School of Information & Telecommunications Eng., Nanjing Univ. of P & T, P. R. China
[5] Dept. of Electronic Engineering, National Taiwan Univ. of Science & Technology, Taiwan, R. O. China
Corresponding author's E-mail : shengbao@ieee.org

arXiv:0707.4289v1 [cs.AI] 29 Jul 2007

***Abstract*—In this paper, we employ Probabilistic Neural Network (PNN) with image and data processing techniques to implement a general purpose automated leaf recognition for plant classification. 12 leaf features are extracted and orthogonalized into 5 principal variables which consist the input vector of the PNN. The PNN is trained by 1800 leaves to classify 32 kinds of plants with an accuracy greater than 90%. Compared with other approaches, our algorithm is an accurate artificial intelligence approach which is fast in execution and easy in implementation.**

***Index Terms*—Probabilistic Neural Network, feature extraction, leaf recognition, plant classification**

I. Introduction

Plants exist everywhere we live, as well as places without us. Many of them carry significant information for the development of human society. The urgent situation is that many plants are at the risk of extinction. So it is very necessary to set up a database for plant protection [1]–[4]. We believe that the first step is to teach a computer how to classify plants.

Compared with other methods, such as cell and molecule biology methods, classification based on leaf image is the first choice for leaf plant classification. Sampling leaves and photoing them are low-cost and convenient. One can easily transfer the leaf image to a computer and a computer can extract features automatically in image processing techniques.

Some systems employ descriptions used by botanists [5]–[8]. But it is not easy to extract and transfer those features to a computer automatically. This paper tries to prevent human interference in feature extraction.

It is also a long discussed topic on how to extract or measure leaf features [9]–[15]. That makes the application of pattern recognition in this field a new challenge [1] [16]. According to [1], data acquisition from living plant automatically by the computer has not been implemented.

Several other approaches used their pre-defined features. *Miao et al.* proposed an evidence-theory-based rose classification [3] based on many features of roses. *Gu et al.* tried leaf recognition using skeleton segmentation by wavelet transform and Gaussian interpolation [17]. *Wang et al.* used a moving median center (MMC) hypersphere classifier [18]. Similar method was proposed by *Du et al.* [1]. Their another paper proposed a modified dynamic programming algorithm for leaf shape matching [19]. *Ye et al.* compared the similarity between features to classify plants [2].

Many approaches above employ k-nearest neighbor (k-NN) classifier [1] [17] [18] while some papers adopted Artificial Neural Network (ANN). *Saitoh et al.* combined flower and leaf information to classify wild flowers [20]. *Heymans et al.* proposed an application of ANN to classify opuntia species [21]. *Du et al.* introduced shape recognition based on radial basis probabilistic neural network which is trained by orthogonal least square algorithm (OLSA) and optimized by recursive OLSA [22]. It performs plant recognition through modified Fourier descriptors of leaf shape.

Previous work have some disadvantages. Some are only applicable to certain species [3] [16] [21]. As expert system, some methods compare the similarity between features [2] [8]. It requires pre-process work of human to enter keys manually. This problem also happens on methods extracting features used by botanists [7] [16].

Among all approaches, ANN has the fastest speed and best accuracy for classification work. [22] indicates that ANN classifiers (MLPN, BPNN, RBFNN and RBPNN) run faster than k-NN (k=1, 4) and MMC hypersphere classifier while ANN classifiers advance other classifiers on accuracy. So this paper adopts an ANN approach.

This paper implements a leaf recognition algorithm using easy-to-extract features and high efficient recognition algorithm. Our main improvements are on feature extraction and the classifier. All features are extracted from digital leaf image. Except one feature, all features can be extracted automatically. 12 features are orthogonalized by Principal Components Analysis (PCA) [23]. As to the classifier, we use PNN [24] for its fast speed and simple structure. The whole algorithm is easy-to-implement, using common approaches.

The rest of this paper is organized as follows. Sec. II discusses image pre-processing. Sec. III introduces how 12 leaf features are extracted. PCA and PNN are discussed in Sec. IV. Experimental results are given in Sec. V. Future work on improving our algorithm is mentioned in Sec. VI. Sec. VII concludes this paper.

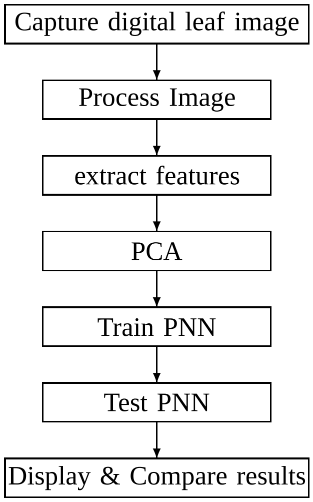

Fig. 1. Flow diagram of proposed scheme

II. Image Pre-processing

A. Converting RGB image to binary image

The leaf image is acquired by scanners or digital cameras. Since we have not found any digitizing device to save the image in a lossless compression format, the image format here is JPEG. All leaf images are in 800 x 600 resolution. There is no restriction on the direction of leaves when photoing.

An RGB image is firstly converted into a grayscale image. Eq. 1 is the formula used to convert RGB value of a pixel into its grayscale value.

$$gray = 0.2989 * R + 0.5870 * G + 0.1140 * B \qquad (1)$$

where R, G, B correspond to the color of the pixel, respectively.

The level to convert grayscale into binary image is determined according to the RGB histogram. We accumulate the pixel values to color R, G, B respectively for 3000 leaves and divide them by 3000, the number of leaves. The average histogram to RGB of 3000 leaf images is shown as Fig. 2.

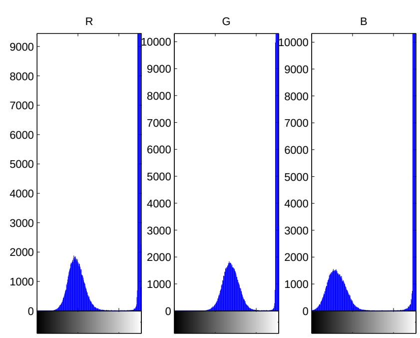

Fig. 2. RGB histogram

There are two peaks in every color's histogram. The left peak refers to pixels consisting the leaf while the right peak refers to pixels consisting the white background. The lowest point between two peaks is around the value 242 on the average. So we choose the level as 0.95 (242/255=0.949). The output image replaces all pixels in the input image with luminance greater than the level by the value 1 and replaces all other pixels by the value 0.

A rectangular averaging filter of size 3×3 is applied to filter noises. Then pixel values are rounded to 0 or 1.

B. Boundary Enhancement

When mentioning the leaf shape, the first thing appears in your mind might be the margin of a leaf. Convolving the image with a Laplacian filter of following 3×3 spatial mask:

$$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array}$$

we can have the margin of the leaf image.

An example of image pre-processing is illustrated in Fig. 3. To make boundary as a black curve on white background, the "0" "1" value of pixels is swapped.

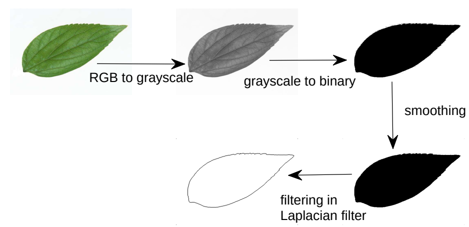

Fig. 3. A pre-processing example

III. Feature Extraction

In this paper, 12 commonly used digital morphological features (DMFs), derived from 5 basic features, are extracted so that a computer can obtain feature values quickly and automatically (only one exception).

A. Basic Geometric Features

Firstly, we obtain 5 basic geometric features.

1) Diameter: The diameter is defined as the longest distance between any two points on the margin of the leaf. It is denoted as D.

2) Physiological Length: The only human interfered part of our algorithm is that you need to mark the two terminals of the main vein of the leaf via mouse click. The distance between the two terminals is defined as the physiological length. It is denoted as L_p.

3) Physiological Width: Drawing a line passing through the two terminals of the main vein, one can plot infinite lines orthogonal to that line. The number of intersection pairs between those lines and the leaf margin is also infinite. The longest distance between points of those intersection pairs is defined at the physiological width. It is denoted as W_p.

Since the coordinates of pixels are discrete, we consider two lines are orthogonal if their degree is $90° \pm 0.5°$.

The relationship between physiological length and physiological width is illustrated in Fig. 4.

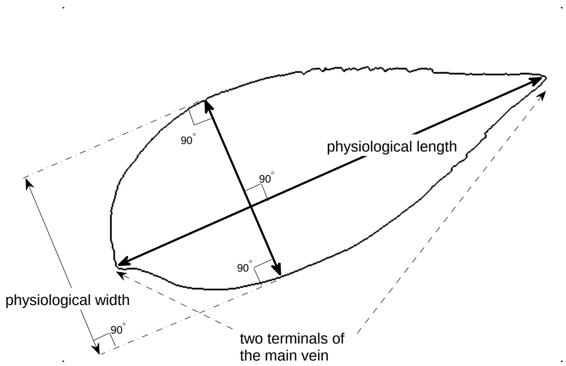

Fig. 4. Relationship between Physiological Length and Physiological Width

4) Leaf Area: The value of leaf area is easy to evaluate, just counting the number of pixels of binary value 1 on smoothed leaf image. It is denoted as A.

5) Leaf Perimeter: Denoted as P, leaf perimeter is calculated by counting the number of pixels consisting leaf margin.

B. 12 Digital Morphological Features

Based on 5 basic features introduced previously, we can define 12 digital morphological features used for leaf recognition.

1) Smooth factor: We use the effect of noises to image area to describe the smoothness of leaf image. In this paper, smooth factor is defined as the ratio between area of leaf image smoothed by 5×5 rectangular averaging filter and the one smoothed by 2×2 rectangular averaging filter.

2) Aspect ratio: The aspect ratio is defined as the ratio of physiological length L_p to physiological width W_p, thus L_p/W_p.

3) Form factor: This feature is used to describe the difference between a leaf and a circle. It is defined as $4\pi A/P^2$, where A is the leaf area and P is the perimeter of the leaf margin.

4) Rectangularity: Rectangularity describes the similarity between a leaf and a rectangle. It is defined as L_pW_p/A, where L_p is the physiological length, W_p is the physiological width and A is the leaf area.

5) Narrow factor: Narrow factor is defined as the ratio of the diameter D and physiological length L_p, thus D/L_p.

6) Perimeter ratio of diameter: Ratio of perimeter to diameter, representing the ratio of leaf perimeter P and leaf diameter D, is calculated by P/D.

7) Perimeter ratio of physiological length and physiological width: This feature is defined as the ratio of leaf perimeter P and the sum of physiological length L_p and physiological width W_p, thus $P/(L_p + W_p)$.

8) Vein features: We perform morphological opening [25] on grayscale image with falt, disk-shaped structuring element of radius 1,2,3,4 and substract remained image by the margin. The results look like the vein. That is why following 5 feature are called vein features. Areas of left pixels are denoted as A_{v1}, A_{v2}, A_{v3} and A_{v4} respectively. Then we obtain the last 5 features: A_{v1}/A, A_{v2}/A, A_{v3}/A, A_{v4}/A, A_{v4}/A_{v1}.

Now we have finished the step of feature acquisition and go on to the data analysis section.

IV. Proposed Scheme

A. Principal Component Analysis (PCA)

To reduce the dimension of input vector of neural network, PCA is used to orthogonalize 12 features. The purpose of PCA is to present the information of original data as the linear combination of certain linear irrelevant variables. Mathematically, PCA transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. Each coordinate is called a principal component.

In this paper, the contribution of first 5 principal components is 93.6%. To balance the computational complexity and accuracy, we adopt 5 principal components.

When using our algorithm, one can use the mapping $f : \mathbb{R}^{12} \rightarrow \mathbb{R}^5$ to obtain the values of components in the new coordinate system.

B. Introduction to Probabilistic Neural Network

An artificial neural network (ANN) is an interconnected group of artificial neurons simulating the thinking process of human brain. One can consider an ANN as a "magical" black box trained to achieve expected intelligent process, against the input and output information stream. Thus, there is no need for a specified algorithm on how to identify different plants. PNN is derived from Radial Basis Function (RBF) Network which is an ANN using RBF. RBF is a bell shape function that scales the variable nonlinearly.

PNN is adopted for it has many advantages [26]. Its training speed is many times faster than a BP network. PNN can approach a Bayes optimal result under certain easily met conditions [24]. Additionally, it is robust to noise examples. We choose it also for its simple structure and training manner.

The most important advantage of PNN is that training is easy and instantaneous [24]. Weights are not "trained" but assigned. Existing weights will never be alternated but only new vectors are inserted into weight matrices when training. So it can be used in real-time. Since the training and running procedure can be implemented by matrix manipulation, the speed of PNN is very fast.

The network classfies input vector into a specific class because that class has the maximum probability to be correct. In this paper, the PNN has three layers: the Input layer, Radial Basis Layer and the Competitive Layer. Radial Basis Layer evaluates vector distances between input vector and row weight vectors in weight matrix. These distances are scaled by Radial Basis Function nonlinearly. Then the Competitive Layer finds the shortest distance among them, and thus finds the training pattern closest to the input pattern based on their distance.

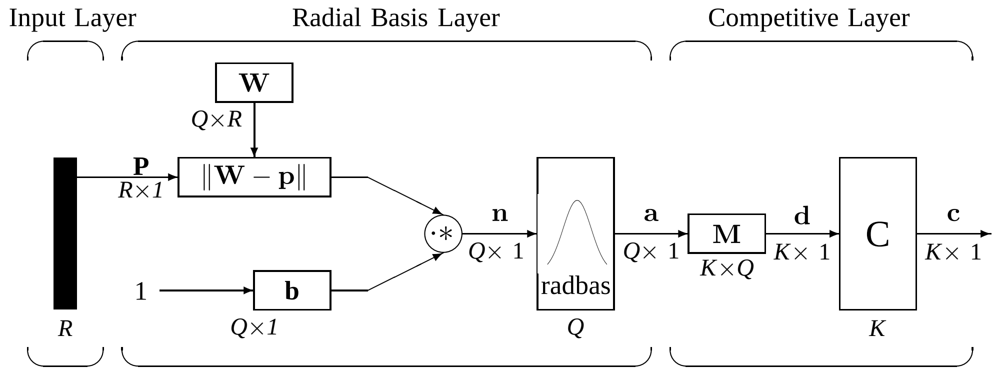

Fig. 5. Network Structure, R=5, Q=1800, K=32

C. Network Structure

The network structure in our purposed scheme is illustrated in Fig. 5. We adopt symbols and notations used in the book *Neural Network Design* [27]. These symbols and notations are also used by MATLAB Neural Network Toolbox [28]. Dimensions of arrays are marked under their names.

1) Input Layer: The input vector, denoted as $\mathbf{p}$, is presented as the black vertical bar in Fig. 5. Its dimension is $R \times 1$. In this paper, $R = 5$.

2) Radial Basis Layer: In Radial Basis Layer, the vector distances between input vector $\mathbf{p}$ and the weight vector made of each row of weight matrix $\mathbf{W}$ are calculated. Here, the vector distance is defined as the dot product between two vectors [29]. Assume the dimension of $\mathbf{W}$ is $Q \times R$. The dot product between $\mathbf{p}$ and the i-th row of $\mathbf{W}$ produces the i-th element of the distance vector $||\mathbf{W} - \mathbf{p}||$, whose dimension is $Q \times 1$, as shown in Fig. 5. The minus symbol, "$-$", indicates that it is the distance between vectors.

Then, the bias vector $\mathbf{b}$ is combined with $||\mathbf{W} - \mathbf{p}||$ by an element-by-element multiplication, represented as "$\cdot *$" in Fig. 5. The result is denoted as $\mathbf{n} = ||\mathbf{W} - \mathbf{p}|| \cdot *\mathbf{p}$.

The transfer function in PNN has built into a distance criterion with respect to a center. In this paper, we define it as

$$radbas(n) = e^{-n^2} \tag{2}$$

Each element of $\mathbf{n}$ is substituted into Eq. 2 and produces corresponding element of $\mathbf{a}$, the output vector of Radial Basis Layer. We can represent the *i*-th element of $\mathbf{a}$ as

$$a_i = radbas(||\mathbf{W}_i - \mathbf{p}|| \cdot *\mathbf{b}_i) \tag{3}$$

where $\mathbf{W}_i$ is the vector made of the i-th row of $\mathbf{W}$ and $\mathbf{b}_i$ is the i-th element of bias vector $\mathbf{b}$.

3) Some characteristics of Radial Basis Layer: The i-th element of $\mathbf{a}$ equals to 1 if the input $\mathbf{p}$ is identical to the i-th row of input weight matrix $\mathbf{W}$. A radial basis neuron with a weight vector close to the input vector $\mathbf{p}$ produces a value near 1 and then its output weights in the competitive layer will pass their values to the competitive function which will be discussed later. It is also possible that several elements of $\mathbf{a}$ are close to 1 since the input pattern is close to several training patterns.

4) Competitive Layer: There is no bias in Competitive Layer. In Competitive Layer, the vector $\mathbf{a}$ is firstly multiplied with layer weight matrix $\mathbf{M}$, producing an output vector $\mathbf{d}$. The competitive function, denoted as $\mathbf{C}$ in Fig. 5, produces a 1 corresponding to the largest element of $\mathbf{d}$, and 0's elsewhere. The output vector of competitive function is denoted as $\mathbf{c}$. The index of 1 in $\mathbf{c}$ is the number of plant that our system can classify. It can be used as the index to look for the scientific name of this plant. The dimension of output vector, K, is 32 in this paper.

D. Network Training

Totally 1800 pure leaves are sampled to train this network. Those leaves are sampled in the campus of Nanjing University and Sun Yat-Sen arboretum, Nanking, China. Most of them are common plants in Yangtze Delta, China. Details about the leaf numbers of different kinds of plants are given in Table I. The reason why we sample different pieces of leaves to different plants is the difficulty to sample leaves varies on plants.

1) Radial Basis Layer Weights: $\mathbf{W}$ is set to the transpose of $R \times Q$ matrix of Q training vectors. Each row of $\mathbf{W}$ consists of 5 principal variables of one trainging samples. Since 1800 samples are used for training, $Q = 1800$ in this paper.

2) Radial Basis Layer Biases: All biases in radial basis layer are all set to $\sqrt{\ln 0.5}/s$ resulting in radial basis functions that cross 0.5 at weighted inputs of $\pm s$. s is called the spread constant of PNN.

The value of s can not be selected arbitrarily. Each neuron in radial basis layer will respond with 0.5 or more to any input vectors within a vector distance of s from their weight vector. A too small s value can result in a solution that does not generalize from the input/target vectors used in the design. In contrast, if the spread constant is large enough, the radial basis neurons will output large values (near 1.0) for all the inputs used to design the network.

In this paper, the s is set to 0.03($\simeq 1/32$) according to our experience.

3) Competitive Layer Weights: $\mathbf{M}$ is set to $K \times Q$ matrix of Q target class vectors. The target class vectors are converted from class indices corresponding to input vectors. This process generates a sparse matrix of vectors, with one 1 in each column, as indicated by indices. For example, if the i-th

TABLE I
DETAILS ABOUT THE LEAF NUMBERS OF DIFFERENT TYPES OF PLANTS

Scientific Name(in Latin)	Common Name	training samples	number of incorrect recognition
Phyllostachys edulis (Carr.) Houz.	pubescent bamboo	58	0
Aesculus chinensis	Chinese horse chestnut	63	0
Berberis anhweiensis Ahnendt	Anhui Barberry	58	0
Cercis chinensis	Chinese redbud	72	1
Indigofera tinctoria L.	true indigo	72	0
Acer Dalmatum	Japanese maple	53	0
Phoebe zhennan S. Lee & F.N. Wei	Nanmu	60	1
Kalopanax septemlobus (Thunb. ex A.Murr.) Koidz	castor aralia	51	0
Cinnamomum japonicum Siebold ex Nakai	Japan Cinnamon	51	2
Koelreuteria paniculata Laxm.	goldenrain tree	57	0
Ilex macrocarpa	holly	50	0
Pittosporum tobira (Thunb.) Ait. f.	Japanese cheesewood	61	1
Chimonanthus praecox L.	wintersweet	51	2
Cinnamomum camphora (L.) J. Presl	camphortree	61	3
Viburnum awabuki	Japanese Viburnum	58	2
Osmanthus fragrans Lour.	sweet osmanthus	55	5
Cedrus deodara (Roxb.) G. Don	deodar	65	3
Ginkgo biloba L.	ginkgo, maidenhair tree	57	0
Lagerstroemia indica (L.) Pers.	Crepe myrtle	57	0
Nerium oleander L.	oleander	61	0
Podocarpus macrophyllus (Thunb.) Sweet	yew plum pine	60	0
Prunus $\times$*yedoensis* Matsumura	Japanese Flowering Cherry	50	0
Ligustrum lucidum Ait. f.	Chinese Privet	52	1
Tonna sinensis M. Roem.	Chinese Toon	58	1
Prunus persica (L.) Batsch	peach	50	2
Manglietia fordiana Oliv.	Ford Woodlotus	50	3
Acer buergerianum Miq.	trident maple	50	1
Mahonia bealei (Fortune) Carr.	Beale's barberry	50	0
Magnolia grandiflora L.	southern magnolia	50	0
Populus $\times$*canadensis* Moench	Carolina poplar	58	3
Liriodendron chinense (Hemsl.) Sarg.	Chinese tulip tree	50	0
Citrus reticulata Blanco	tangerine	51	0

sample in training set is the j-th kind of plant, then we have one 1 on the j-th row of i-th column of $\mathbf{M}$.

V. EXPERIMENTAL RESULT

To each kind of plant, 10 pieces of leaves from testing sets are used to test the accuracy of our algorithm. Numbers incorrect recognition are listed in the last column of Table I. The average accuracy is 90.312%.

Some species get a low accuracy in Table I. Due to the simplicity of our algorithm framework, we can add more features to boost the accuracy.

We compared the accuracy of our algorithm with other general purpose (not only applicable to certain species) classification algorithms that only use leaf-shape information. According to Table II, the accuracy of our algorithm is very similar to other schemes. Considering our advantage respect to other automated/semi-automated general purpose schemes, easy-to-implement framework and fast speed of PNN, the performance is very good.

The source code in MATLAB can be downloaded now from http://flavia.sf.net.

TABLE II
ACCURACY COMPARISON

Scheme	Accuracy
proposed in [2]	71%
1-NN in [17]	93%
k-NN ($k = 5$) in [17]	86%
RBPNN in [17]	91%
MMC in [1]	91%
k-NN ($k = 4$) in [1]	92%
MMC in [18]	92%
BPNN in [18]	92%
RBFNN in [22]	94%
MLNN in [22]	94 %
Our algorithm	90%

VI. FUTURE WORK

Since the essential of the competitive function is to output the index of the maximum value in an array, we plan to let our algorithm output not only the index of maximum value, but also the indices of the second greatest value and the third greatest value. It is based on this consideration that the index of the second greatest value corresponds to the second top matched plant. So does the index of the third greatest value. Sometimes, maybe the correct plant is in the second or the third most possible plant. We are going to provide all these three possible answers to users. Further more, users can choose the correct one they think so that our algorithm can learn from it to improve its accuracy.

Other features are also under consideration. Daniel Drucker from Department of Psychology, University of Pennsylvania, suggested us to use Fourier Descriptors so that we can do some mathematical manipulations later. We are also trying to use other features having psychology proof that is useful for human to recognize things like the leaf, such as the surface qualities [30].

Our plant database is under construction. The number of plants that can be classified will be increased.

VII. Conclusion

This paper introduces a neural network approach for plant leaf recognition. The computer can automatically classify 32 kinds of plants via the leaf images loaded from digital cameras or scanners. PNN is adopted for it has fast speed on training and simple structure. 12 features are extracted and processed by PCA to form the input vector of PNN. Experimental result indicates that our algorithm is workable with an accuracy greater than 90% on 32 kinds of plants. Compared with other methods, this algorithm is fast in execution, efficient in recognition and easy in implementation. Future work is under consideration to improve it.

Acknowledgements

Prof. Xin-Jun Tian, Department of Botany, School of Life Sciences, Nanjing University provided the lab and some advises for this research.

Yue Zhu, a master student of Department of Botany, School of Life Sciences, Nanjing University, helped us sampling plant leaves.

Ang Li and Bing Chen from Institute of Botany, Chinese Academy of Science, provided us some advises on plant taxonomy and searched the scientific name for plants.

Shi Chen, a PhD student from School of Agriculture, Pennsylvania State University, initiated another project which inspired us this research.

The authors also wish to thank secretary Crystal Hwan-Ming Chan, for her assistance to our project.

References

[1] J.-X. Du, X.-F. Wang, and G.-J. Zhang, "Leaf shape based plant species recognition," *Applied Mathematics and Computation*, vol. 185, 2007.

[2] Y. Ye, C. Chen, C.-T. Li, H. Fu, and Z. Chi, "A computerized plant species recognition system," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, October 2004.

[3] Z. Miao, M.-H. Gandelin, and B. Yuan, "An oopr-based rose variety recognition system," *Engineering Applications of Artificial Intelligence*, vol. 19, 2006.

[4] R. de Oliveira Plotze, M. Falvo, J. G. Pdua, L. C. Bernacci, M. L. C. Vieira, G. C. X. Oliveira, and O. M. Bruno, "Leaf shape analysis using the multiscale minkowski fractal dimension, a new morphometric method: a study with passifliora (passifloraceae)," *Canada Journal of Botany*, vol. 83, 2005.

[5] M. J. Dallwitz, "A general system for coding taxonomic descriptions," *Taxon*, vol. 29, 1980.

[6] H. Fu, Z. Chi, D. Feng, and J. Song, "Machine learning techniques for ontology-based leaf classification," in *IEEE 2004 8th International Conference on Control, Automation, Robotics and Vision*, Kunming, China, 2004.

[7] D. Warren, "Automated leaf shape description for variety testing in chrysanthemums," in *Proceedings of IEE 6th International Conference Image Processing and Its Applications*, 1997.

[8] T. Brendel, J. Schwanke, P. Jensch, and R. Megnet, "Knowledge-based object recognition for different morphological classes of plants," *Proceedings of SPIE*, vol. 2345, 1995.

[9] Y. Li, Q. Zhu, Y. Cao, and C. Wang, "A leaf vein extraction method based on snakes technique," in *Proceedings of IEEE International Conference on Neural Networks and Brain*, 2005.

[10] H. Fu and Z. Chi, "Combined thresholding and neural network approach for vein pattern extraction from leaf images," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 153, no. 6, December 2006.

[11] Y. Nam, E. Hwang, and K. Byeon, "Elis: An efficient leaf image retrieval system," in *Proceedings of International Conference on Advances in Pattern Recognition 2005*, ser. LNCS 3687. Springer, 2005.

[12] H. Fu and Z. Chi, "A two-stage approach for leaf vein extraction," in *Proceedings of IEEE International Conference on Neural Networks and Signal Processing*, Nanjing, China, 2003.

[13] Z. Wang, Z. Chi, and D. Feng, "Shape based leaf image retrieval," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 1, February 2003.

[14] H. QI and J.-G. YANG, "Sawtooth feature extraction of leaf edge based on support vector machine," in *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, November 2003.

[15] S. M. Hong, B. Simpson, and G. V. G. Baranoski, "Interactive venation-based leaf shape modeling," *Computer Animation and Virtual Worlds*, vol. 16, 2005.

[16] F. Gouveia, V. Filipe, M. Reis, C. Couto, and J. Bulas-Cruz, "Biometry: the characterisation of chestnut-tree leaves using computer vision," in *Proceedings of IEEE International Symposium on Industrial Electronics*, Guimarães, Portugal, 1997.

[17] X. Gu, J.-X. Du, and X.-F. Wang, "Leaf recognition based on the combination of wavelet transform and gaussian interpolation," in *Proceedings of International Conference on Intelligent Computing 2005*, ser. LNCS 3644. Springer, 2005.

[18] X.-F. Wang, J.-X. Du, and G.-J. Zhang, "Recognition of leaf images based on shape features using a hypersphere classifier," in *Proceedings of International Conference on Intelligent Computing 2005*, ser. LNCS 3644. Springer, 2005.

[19] J.-X. Du, D.-S. Huang, X.-F. Wang, and X. Gu, "Computer-aided plant species identification (capsi) based on leaf shape matching technique," *Transactions of the Institute of Measurement and Control*, vol. 28, 2006.

[20] T. K. Takeshi Saitoh, "Automatic recognition of wild flowers," in *Proceedings of 15th International Conference on Pattern Recognition (ICPR'00)*, vol. 2, 2000.

[21] B. C. Heymans, J. P. Onema, and J. O. Kuti, "A neural network for opuntia leaf-form recognition," in *Proceedings of IEEE International Joint Conference on Neural Networks*, 1991.

[22] J. Du, D. Huang, X. Wang, and X. Gu, "Shape recognition based on radial basis probabilistic neural network and application to plant species identification," in *Proceedings of 2005 International Symposium of Neural Networks*, ser. LNCS 3497. Springer, 2005.

[23] J. Shlens. (2005, December) A tutorial on principal component analysis. [Online]. Available: http://www.cs.cmu.edu/∼elaw/papers/pca.pdf

[24] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, 1990.

[25] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Prentice Hall, 2004.

[26] T. Master, *Practical Neural Network Recipes*. New York: John Wiley, 1993.

[27] M. T. Hagan, H. B. Demut, and M. H. Beale, *Neural Network Design*, 2002.

[28] (2007) Matlab neural network toolbox documentation. MathWorks. Inc. [Online]. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/radial10.html#8378

[29] D. F. Specht, "Probabilistic neural networks for classification mapping, or associative memory," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 1, 1988.

[30] I. Motoyoshi, S. Nishida, L. Sharan, and E. H. Adelson, "Image statistics and the perception of surface qualities," *Nature*, vol. 447, May 2007.