Academic Journal of Computing & Information Science
ISSN 2616-5775 Vol. 2, Issue 3: 68-74, DOI: 10.25236/AJCIS.020308

Leaf Contour Recognition Using Hash Learning

Shuman Pan, Tao Li, Yichen Yin

College of Science, Beijing Forestry University, Beijing 100083, China

ABSTRACT. *In the era of big data, hash learning has gained wide attention in large-scale images classification and recognition. In this paper, we identify the type of leaves by hash learning. Firstly, we process original leaf images into contour images and calculate the values of eight geometric feature for each image. Then, we set optimal thresholds for each feature and perform hash mapping. Finally, the type of leaves is determined by calculating the similarity of their hash codes. We conduct an experiment on 14 types of leaves and use one deep learning model as a comparative experiment. The results show that our method has higher recognition accuracy and recognition efficiency as well as reduces the overhead of data storage and transmission.*

KEYWORDS: *leaf recognition, image processing, shape features, hash learning.*

1. Introduction

Plant is the most diverse and widely distributed creature on Earth. It is not only a necessary resource for human production and life, but also a key element for maintaining ecological balance. It can be seen that researches on plant classification and recognition can create vital value for society. In general, one plant has six organs: root, stem, leaf, flower, fruit, and seed. Relatively speaking, the leaves have more advantages such as convenient collection, stable state, and large differences in visual types, which provide important clues [1] for scholars at home and abroad. However, with the wave of the "big data era", the field of plant image recognition, like other fields, faces the problem of "massive data" and "dimensional disaster" [2], and the efficiency of the original nearest neighbor search algorithm is reduced or even invalid.

Therefore, the machine learning community has been trying to learn "new feature data", and the approximate nearest neighbor search algorithms are becoming more and more important. Among them, hash learning is a research hotspot in recent years. It maps data into a binary string through a machine learning mechanism, and meanwhile makes the hash code maintain the neighbor relationship in the original space as much as possible [3], that is, two similar pictures in the original space are mapped into two similar strings in the Hamming space. In this way, hash learning can not only significantly reduce the data storage and transmission overhead, but also reduce the data dimension, thereby significantly improving the efficiency of the big data learning system.

Based on the traditional work of collecting leaf images and extracting leaf shape features, this paper introduces hash learning to realize rapid and successful identification of the type of plants.

2. Blade Outline Description

2.1 Data

This paper selects 14 types of leaves from the Flavia dataset [4] and each with 50-72 samples. Figure 1 shows a sample plot of each type of the blades.

Figure 1: A sample plot of each type of the blades

2.2 Image preprocessing

As shown in Figure 2, we process the original blade images into contour images by the following steps.

1) Low-pass filtering removes noise points in the image.

2) Flooding fill algorithm fixes small holes in the image.

3) Gray processing turns a color image into a grayscale image.

4) Opening-and-closing operations further removes isolated points, burrs, bridges, and repair holes.

5) Use the OTUS method [5] to convert the grayscale image to a binary image.

6) Detect the outline and present the outline pixels as a line to generate a contour image.

ISSN 2616-5775 Vol. 2, Issue 3: 68-74, DOI: 10.25236/AJCIS.020308

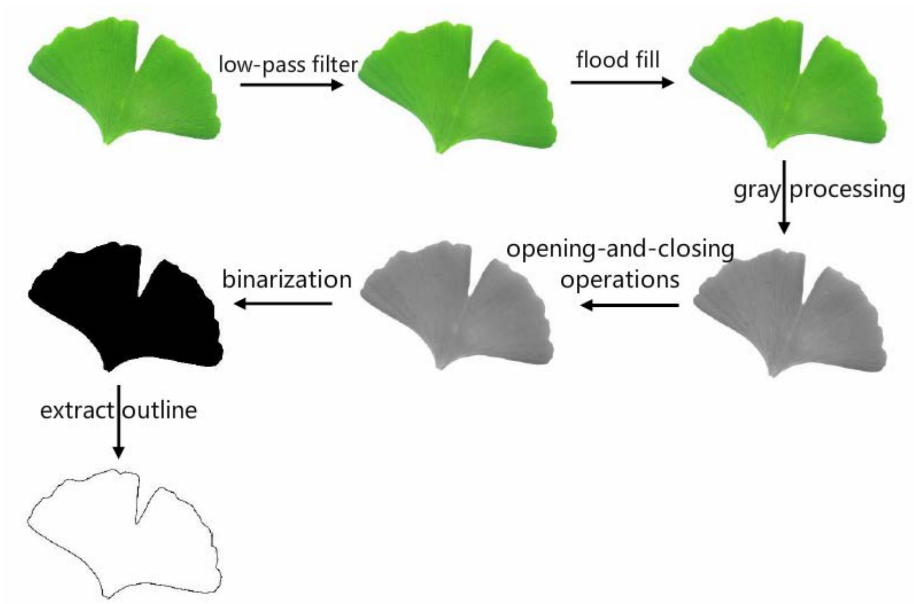

Figure 2: preprocessing steps

2.3 Shape descriptions

Different types of blade shapes are different, and their shape features are important clues for human vision and computer vision to distinguish the type [6]. In this paper, the contour line is described by calculating Aspect Ratio, Rectangularity, Circularity, Perimeter Convexity, Area Convexity, Sphericity, Form Factor and Eccentricity. These eight features have rotation, translation and affine invariance [7].

3. Hash map

3.1 Hash function

After calculating the values of eight geometric features, we set optimal thresholds for each feature by the threshold hash method (Algorithm 1) to realize the maximum discrimination for the blades.

ISSN 2616-5775 Vol. 2, Issue 3: 68-74, DOI: 10.25236/AJCIS.020308

Algorithm 1: the algorithm of the threshold hash method

Input: The training set
Output: Eight thresholds
Step1: Calculate the values of eight geometric features of each sample leaf
Step2: For each geometric feature, sort the values from small to large, define the deviation value of this leaf

$$d_j = min(n_{ij}, m_{ij}),\ i=1, 2, \ldots, 8, j=1, 2, \ldots, 14.$$

where n_{ij} is the number of the i-th eigenvalue of the j-th leaf less than the threshold and m_{ij} is the number of the i-th eigenvalue of the j-th leaf greater than the threshold.
Step3: If the value range of each geometric feature value is regarded as [0, 1], traverse the point in the interval of [0.4, 0.6], and calculate the average deviation of 14 leaves from this point. The point at which the average deviation value is the smallest is the threshold.

The hash mapping function is as follows

$$\varphi(\text{hash\_code}_i) = \begin{cases} 1, & \text{value}_i \geq \text{threshold}_i \\ 0, & \text{value}_i < \text{threshold}_i \end{cases} \quad (1)$$

Table 1 corresponding hash codes of one Ginkgo biloba leaf

Geometric Feature	Threshold	ginkgo biloba	Hash Code
Aspect Ration	1.921312	1.492617	0
Rectangularity	0.67444	0.63568	0
Circularity	0.946938	0.283158	0
Perimeter Convexity	1.08511	1.238157	1
Area Convexity	0.233291	0.869573	1
Sphericity	0.708978	0.741415	1
Form Factor	0.446409	0.483626	1
Eccentricity	1.972141	1.772141	0
Hash Code	\	\	00011110

Calculate the average deviation of 14 leaves from this point. The point at which the average deviation value is the smallest is the threshold. Table 1 shows an example of hash code of a Ginkgo biloba leaf.

ISSN 2616-5775 Vol. 2, Issue 3: 68-74, DOI: 10.25236/AJCIS.020308

3.2 Similarity measure

After hash mapping, calculating the similarity of two leaf images is transformed into calculating the similarity of the corresponding hash codes, that is, the probability that their hash values are equal.

$$Pr\left(h(x_i) = h(x_j)\right) = sim(x_i, x_j) \qquad (2)$$

Among them, *sim*(·) is a similarity measure function, *Pr*(·) is a probability, and h is a hash function [8]. Finally, the blade type is determined based on the calculated similarity.

4. Experimental results

4.1 Evaluation indexes

In this paper, the average recognition accuracy (MAP) [9]

$$\text{MAP} = \frac{1}{14}\sum_{\text{i}=1}^{14}\frac{n_i}{N_i}, \qquad (3)$$

and average recognition time (ART) are used to evaluate the recognition results.

$$\text{ART} = \frac{1}{M}\sum_{\text{j}=1}^{M} t_j \qquad (4)$$

4.2 Experimental results

In this paper, the leaves are hashed and identified according to the above method. Its average recognition accuracy is 80.5%, and its average recognition time is 0.1 seconds. Besides, the recognition accuracy of six types of leaves is greater than 90%. Then, we use one deep learning model to conduct a comparative experiment.

The deep learning model mainly uses a convolutional neural network to identify and classify the target image (Figure 3). Firstly, in order to prevent over-fitting, we divide the pre-processed contour dataset into training sets and test sets, and then the size of each image is uniformly converted into a single-channel image with the size of 128×128. The images are then batched into the convolutional neural network.

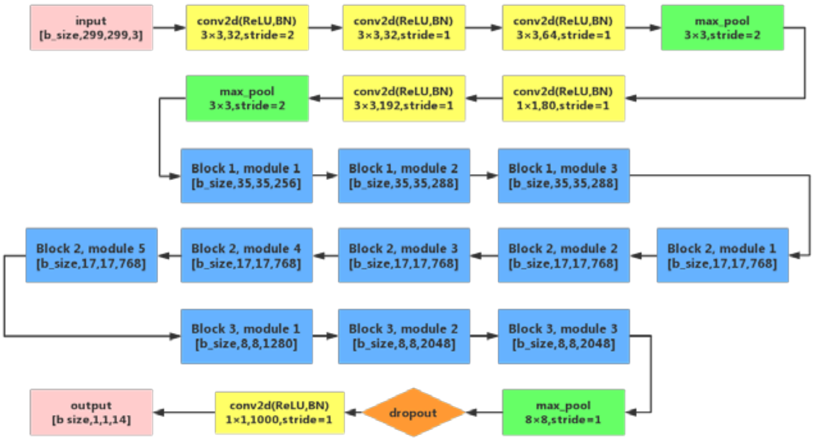

Figure 3: Deep learning model structure

In each convolution layer, we add the BN layer (batch normalization) to speed up the convergence of the neural network and improve the accuracy of its recognition. The learning rate is 0.001, the learning rate attenuation factor is 0.76, and the number of images per batch is 32. After 10,000 steps of training, the loss function tends to be stable. Finally, the test set is input into the neural network. Its average recognition accuracy is 96.6% and its average recognition time is 8.5 seconds. The performance is listed in Table 2.

Table 2: Recognition effect comparison table

Index	ART	MAP
Hash Learning	0.1s	80.5%
Deep Learning	8.5s	96.6%

5. Conclusions

It can be seen from the experimental results that hash learning has good recognition accuracy. Although use contour images to identify has the risk of obtaining inaccurate results due to the lack of features and error superposition, the advantages of contour images in data transfer efficiency, data storage overhead are obvious. In such cases, the advantage of deep learning is limited, although its average recognition accuracy is higher, but its recognition time is too long. If we can find a better way to describe the shape, find a better threshold determination method or find a better hash function, hash learning can ensure high recognition accuracy while ensuring recognition efficiency.

Acknowledgments

This work is supported by the Training Programs of Innovation and Entrepreneurship for Undergraduates No. S201810022092.

References

[1] CHEN Liangxi, WANG Bin. Comparative study of leaf image recognition algorithms based on shape features[J]. Computer Engineering and Applications, 2017, 53(9):17-25.
[2] Hua Qiang, Guo Xinxin, Zhang Feng, et al. Hash retrieval algorithm based on random forest[J]. Computer Science and Exploration, 2019(7):1174-1183.
[3] Li Wujun, Zhou Zhihua. Big Data Hash Learning: Current Situation and Trends [J]. Science Bulletin, 2015, 60(Z1): 485-490.
[4] SG Wu, FS Bao, EY Xu, Y. Wang, Y. Chang and Q. Xiang. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network [M]. 2007 IEEE International Symposium on Signal Processing and Information Technology, Giza, 2007, pp. 11-16.
[5] Moghaddam, Reza Farrahi, and M. Cheriet. AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization[J]. Pattern Recognition, 2012, 45(6):2419-2431.
[6] ZHU Ji-chao, LU Peng. A large-scale image retrieval method based on shape features[J]. Software, 2012, 33(12): 299-304.
[7] WANG Xiaofeng, HUANG Deshuang, DU Jixiang, ZHANG Guojun.Study on feature extraction and recognition of leaf image[J].Computer Engineering and Applications, 2006(03):190-193.
[8] Yuan Peisen, Zhang Yong, Li Meiling, et al. Research on trademark image retrieval based on deep hash learning[J]. Journal of East China Normal University(Natural Science), 2018, 201(05): 180-190.
[9] Zuo Xin, Shen Jifeng, Yu Hualong, Gao Shang, Xu Dan, Hu Chunlong.Image retrieval method based on hash coding learning[J]. Journal of Jiangsu University of Science and Technology (Natural Science Edition), 2015, 29(6): 567 -573.