

Chapter 38

Leaf Classification Methods Based on SVM and SIFT

Yida Ye

Abstract In this research, Support Vector Machine (SVM) and Scale-Invariant Feature Transform (SIFT) are used to identify plants. For each leaf image, the algorithm localizes the keypoints and assigns orientations for each keypoint. Then it matches the sample leaves with the comparison leaves to find out whether they belong to the same category. After conducting edge detection and feature extraction, the experimental result shows that the method for classification gives average accuracy of approximately 99 % when it is tested on 12 descriptive features.

Keywords Leaf classification • SVM • SIFT • Edge detection • Feature extraction

38.1 Introduction

Trees can be seen everywhere in our daily life, but seldom do we observe trees carefully. Leaves vary a lot from shape to size and have some interesting specialities such as phototropism [1]. In this paper some features of leaves are first discussed and then several models to classify leaves and weigh leaves are built. Finally, the strength and weakness of our methods and propose some improvements are analyzed.

Y. Ye (✉)

School of Electrical and Electronic Engineering, North China Electric Power University,
2 Beinong Road, Huilongguan Town, Changping District, Beijing, China
e-mail: sonata1000years@sina.com

38.2 Leaf Classification

38.2.1 Leaf Classification Based on SVM

38.2.1.1 Feature Extraction

Twelve commonly used digital morphological features, derived from five basic features, are extracted so that a computer or smart phone can obtain feature values quickly and automatically.

Basic Geometric Features

- **Diameter:** The diameter is defined as the longest distance between any two points on the margin of the leaf. It is denoted as D .
- **Physiological length:** The distance between the two terminals of the main veins of the leaf is defined as the physiological length. It is denoted as L_p .
- **Physiological width:** The maximum length of a line, which is orthogonal to the main vein, is defined as the physiological width. It is denoted as W_p . Since the coordinates of pixels are discrete, two lines are orthogonal if their degree is $90 \pm 0.5^\circ$. The relationship between Physiological Length and Physiological Width is shown in Fig. 38.1.
- **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 .
- **Leaf perimeter:** Denoted as P , leaf perimeter is calculated by counting the number of pixels consisting leaf margin.

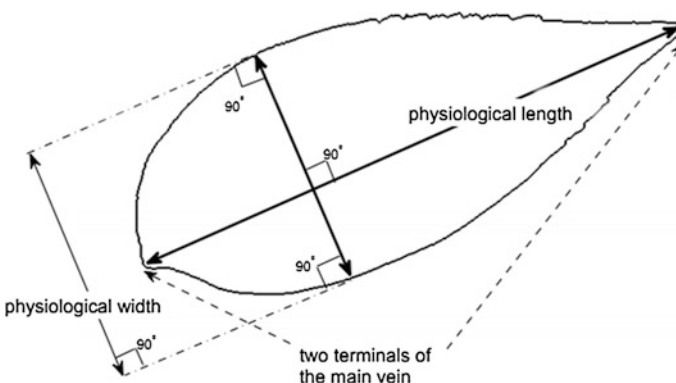


Fig. 38.1 The relationship between L_p and W_p

Digital Morphological Features

- **Smooth factor:** 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 .
- **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.
- **Rectangularity:** Rectangularity describes the similarity between a leaf and a rectangle. It is defined as $L_p W_p/A$, where L_p is the physiological length, W_p is the physiological width and A is the leaf area.
- **Narrow factor:** Narrow factor is defined as the ratio of the diameter D and physiological length L_p , thus D/L_p .
- **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 .
- **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 $P/(L_p + W_p)$.

38.2.1.2 Support Vector Machine

The support vector machine is currently the most popular approach for supervised learning.

SVM finds the OSH (optimum separation hyperplane) by maximizing the margin between the classes [3]. The main concepts of SVM are to first transform input data into a higher dimensional space by means of a kernel function and then construct an OSH between the two classes in the transformed space. Those data vectors nearest to the constructed line in the transformed space are called the support vectors. The SVM estimates a function for classifying data into two classes. Using a nonlinear transformation that depends on a regularization parameter, the input vectors are placed into a high-dimensional feature space, where a linear separation is employed.

To construct a nonlinear support vector classifier, the inner product (x,y) is replaced by a kernel function $K(x,y)$

$$f(x) = \text{sgn} \left(\sum_{i=1}^l \alpha_i y_i K(x_i, x) + b \right)$$

where $f(x)$ determines the membership of x . In this study, the normal subjects were labeled as -1 and other subjects as $+1$. The SVM has two layers. During the

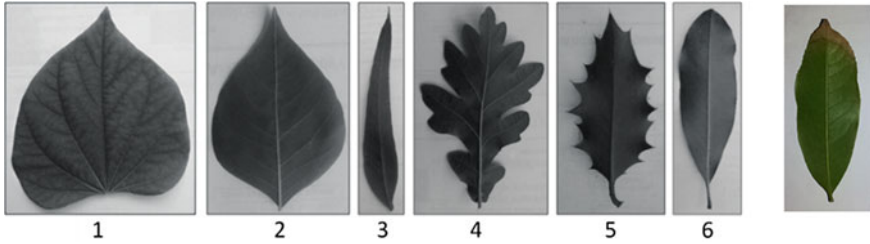


Fig. 38.2 Leaf class and sample leaf

learning process, the first layer selects the basis $K(xi, x)$, $i = 1, 2, \dots, N$ from the given set of kernels, while the second layer constructs a linear function in the space. This is equivalent to finding the optimal hyper plane in the corresponding feature space [4, 5]. The SVM algorithm can construct a variety of learning machines using different kernel functions. Based on 100 trials, the result of classification is approximately 99 % accuracy (Fig. 38.2).

38.2.2 Leaf Classification Based on SIFT

38.2.2.1 SIFT Algorithm

Scale-Space Extrema Detection

This is the stage where the interest points, which are called keypoints in the SIFT framework, are detected. For this, the image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images are taken. Keypoints are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales [6]. Specifically, a DoG image $D(x, y, \sigma)$ is given by

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$$

where $L(x, y, \sigma)$ is the convolution of the original image $I(x, y)$ with the Gaussian blur $G(x, y, \sigma)$ at scale $k\sigma$, i.e.,

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$$

Keypoint Localization

Scale-space extrema detection produces too many keypoint candidates, some of which are unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures. This

information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

Interpolation of Nearby Data for Accurate Position

First, for each candidate keypoint, interpolation of nearby data is used to accurately determine its position. The initial approach was to just locate each keypoint at the location and scale of the candidate keypoint. The new approach calculates the interpolated location of the extremum, which substantially improves matching and stability. The interpolation is done using the quadratic Taylor expansion of the Difference-of-Gaussian scale-space function, $D(x, y, \sigma)$ with the candidate keypoint as the origin. This Taylor expansion [6] is given by:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$

where D and its derivatives are evaluated at the candidate keypoint and $x = (x, y, \sigma)$ is the offset from this point. The location of the extremum, \hat{x} , is determined by taking the derivative of this function with respect to x and setting it to zero. If the offset \hat{x} is larger than 0.5 in any dimension, then that's an indication that the extremum lies closer to another candidate keypoint.

Eliminating Edge Responses

For poorly defined peaks in the DoG function, the principal curvature across the edge would be much larger than the principal curvature along it. Finding these principal curvatures amounts to solving for the eigenvalues of the second-order Hessian matrix, \mathbf{H} :

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The eigenvalues of \mathbf{H} are proportional to the principal curvatures of D . It turns out that the ratio of the two eigenvalues, say α is the larger one, and β the smaller one, with ratio $r = \alpha/\beta$, is sufficient for SIFT's purposes. The trace of \mathbf{H} , i.e., $D_{xx} + D_{yy}$, gives us the sum of the two eigenvalues, while its determinant, i.e., $D_{xx}D_{yy} - D_{xy}^2$, yields the product. The ratio can be shown to be equal to, which depends only on the ratio of the eigenvalues rather than their individual values.

Orientation Assignment

In this step, each keypoint is assigned one or more orientations based on local image gradient directions. This is key step in achieving invariance to rotation as

the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation. The Gaussian-smoothed image $L(x, y, \sigma)$ at the keypoint's scale σ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L(x, y)$ at scale σ , the gradient magnitude $m(x, y)$, and orientation $\theta(x, y)$, are precomputed using pixel differences:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)$$

Keypoint Descriptor

It is important to compute a descriptor vector for each keypoint such that the descriptor is highly distinctive and partially invariant to the remaining variations such as illumination, 3D viewpoint, etc. First a set of orientation histograms are created on 4×4 pixel neighborhoods with eight bins each. These histograms are computed from magnitude and orientation values of samples in a 16×16 region around the keypoint such that each histogram contains samples from a 4×4 subregion of the original neighborhood region. The magnitudes are further weighted by a Gaussian function with σ equal to one half the width of the descriptor window. The descriptor then becomes a vector of all the values of these histograms. Since there are $4 \times 4 = 16$ histograms each with eight bins the vector has 128 elements. This vector is then normalized to unit length in order to enhance invariance to affine changes in illumination. To reduce the effects of non-linear illumination a threshold of 0.2 is applied and the vector is again normalized (Figs. 38.3, 38.4).

It can clearly draw from both graphs that the heads and tails are matched perfectly and several key points on the margin are also matched [7]. It is difficult to get the optimized match in the absence of a large database on the a leaf shape. However, the effectivity of the SIFT algorithm in leaf classification is quite obvious and can be applied better [8] (Figs. 38.5, 38.6).

Fig. 38.3 72 key points

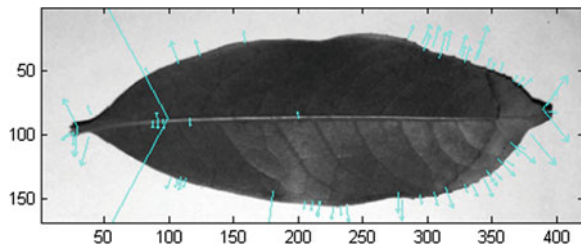


Fig. 38.4 85 key points

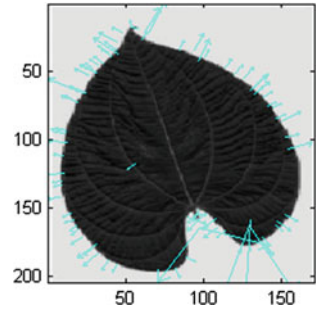


Fig. 38.5 11 matches

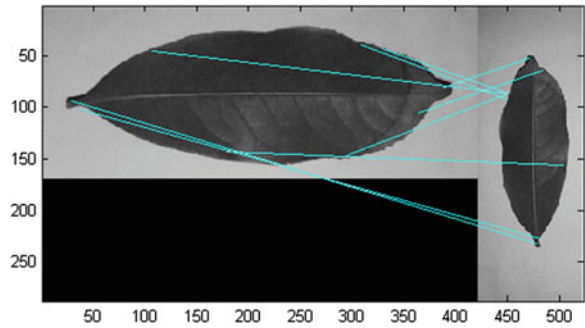
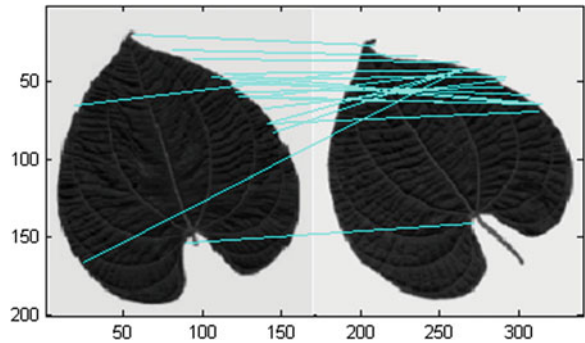


Fig. 38.6 20 matches



38.3 Summary

Two models are built to automatically classify leaves. Model one is based on SVM and edge detection is conducted to get a compact and abstract representation of a leaf. Then six typical descriptive features of leaves are applied to train SVM and the accuracy of classification is approximately 99 %. Model two is based on SIFT. The algorithm localizes the keypoints and assigns orientations for each keypoint. The leaves' heads and tails and keypoints along the margins are perfectly matched. The classification will be more accurate if based on a larger comparison group.

References

1. Kadir A et al (2011) Leaf classification using shape, color, and texture features. *Int J Comput Trends Technol*, July to Aug Issue: 225–230
2. Wu SG et al (2007) A leaf recognition algorithm for plant classification using probabilistic neural network. In: *IEEE international symposium on signal processing and information technology*, IEEE
3. Beghin T, Cope JS, Remagnino P et al (2010) Shape and texture based plant leaf classification. In: *Advanced concepts for intelligent vision systems*, Springer, pp 345–353
4. Singh K, Gupta I, Gupta S (2010) SVM-BDT PNN and fourier moment technique for classification of leaf shape. *Int J Sig Proc Image Proc Pattern Recogn* 3(4)
5. Knight D, Painter J, Potter M, Automatic plant leaf classification for a mobile field guide
6. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
7. Tzionas P, Papadakis SE, Manolakis D (2005) Plant leaves classification based on morphological features and a fuzzy surface selection technique. In: *5th international conference technology and automation*, Thessaloniki, Greece, pp 365–370
8. Wang D, Zhang X, Liu Y (2006) Recognition system of leaf images based on neuronal network. *J For Res* 17(3):243–246