IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 29, 2020, accepted November 9, 2020, date of publication November 16, 2020, date of current version December 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037649

Fast Plant Leaf Recognition Using Improved Multiscale Triangle Representation and KNN for Optimization

JIANYU SU[1], MEIHUA WANG[2], ZHENXIN WU[1], AND QINGLIANG CHEN[1,3]
[1]Department of Computer Science, Jinan University, Guangzhou 510632, China
[2]College of Mathematics and Informatics, South China Agricultural of University, Guangzhou 510642, China
[3]Guangzhou Xuanyuan Research Institute Company, Ltd., Guangzhou 510006, China

Corresponding author: Meihua Wang (wangmeihua@scau.edu.cn)

This work was supported in part by the Basic and Applied Basic Research Fund of Guangdong Province under Grant 2019B1515210009 and in part by the National Natural Science Foundation of China under Grant 62071201 and Grant 61976052.

ABSTRACT Due to the complexity and similarity of plant leaves, it is very important to study an effective leaf-feature extraction method to improve the recognition rate of plant leaves. We study five multi-scale triangle representations: the triangle unsigned area representation (TUA), the triangle vertex angle representation (TVA) and three new representations, which we define as the gray average (TGA), the gray standard deviation (TGSD) and the side length integral (TSLI) on the triangle. In this method the curvature features of the contour, the texture features and the shape area feature are extracted to provide a multiscale leaf-feature description, and a new adaptive KNN for optimization method is proposed to improve the retrieval rate of leaf datasets. Experiments show that compared with the state-of-the-art methods, our method has higher accuracy on the Swedish and Flavia plant leaf datasets, which are respectively 99.35% and 99.43% with 84.76% Mean Average Precision (MAP) value and has comparable results on MPEG-7, kimia99 and kimia216 datasets. When our method is combined with KNN for optimization, the retrieval rate of the above datasets has been significantly improved, especially MAP on the Flavia dataset increases to 94.48%.

INDEX TERMS Plant leaf recognition, multi-scale leaf-feature description, multi-scale triangle representation, adaptive KNN for optimization.

I. INTRODUCTION

The taxonomic investigation of plants is of great significance to the protection of biodiversity, agricultural ecology, and biotechnology safety, and is an important research topic in the field of biology and environmental science. Fast and accurate automatic plant identification not only has a great effect on the classification and investigation of plants, but also has important significance for maintaining biodiversity [3].

Traditional artificial plant identification methods require operators to have a certain degree of professional knowledge and need to know a variety of plants. In addition, traditional plant identification methods often have problems such as large workload, low work efficiency, the operators are easily affected by subjective factors. Therefore, the introduction of automatic plant recognition technology into plant classification can greatly improve the efficiency of classification investigation. More importantly, automatic plant recognition technology has the advantages of simple operation, and it does not require operators to have deep plant professional knowledge, and can also assist experienced botanists and plant ecologists to a certain extent [11]. The recognition of plant species [26] usually involves the observation of certain morphological characteristics of plants, and leaves are important vegetative organs of plants. The leaves of different plants have certain differences, so it is a very common method to use leaves to identify plants.

Automatic leaf recognition technology involves the retrieval and classification of leaf images. Many researchers are interested in extracting multi-scale contour features [1], [5]–[8], [13] from leaf images, they believe that contour features can universally describe the overall layout and local details of the leaf; while there are many researchers interested in shape area features [10], [14], which are calculated on the shapes fitted by the contour points of the leaf images, and a series of parameters such as aspect ratio, area, centroid and eccentricity; leaf recognition and retrieval technology based on texture features [9], [20] extracts the gray-level

The associate editor coordinating the review of this manuscript and approving it for publication was Guitao Cao.

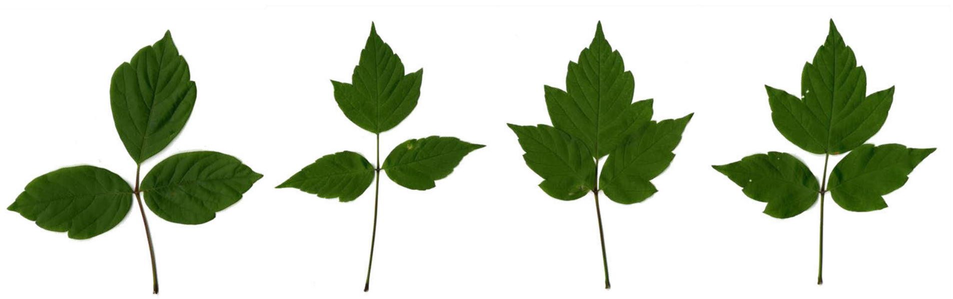

FIGURE 1. Plant samples of the same type.

co-occurrence matrix, Local Binary Patterns and Gabor features of the leaf image. Whether above methods are based on leaf contour features, leaf shape area features or leaf texture features, they are all of great significance to leaf retrieval and classification. Nowadays, combining the above three features is considered a very effective leaf retrieval and classification method [2], [12], among which [12] gets very high accuracy by using the ELM classifiers. An effective leaf image retrieval and classification algorithm usually has the following characteristics:

1) Plant databases are getting larger and larger [6], and searching for similar plants from large-scale databases requires low computational cost.
2) The demand for mobile devices and microcomputers is rising, and the leaf retrieval and classification program needs more lightweight and low-consumption algorithms (such as [6], [7]).
3) The leaves of the same type of plants are quite different (see Fig. 1), which brings great difficulties to accurate retrieval.

In order to solve the above difficulties, we propose a new multi-scale leaf-feature description. The main contributions of this paper are summarized as below:

1) Propose a multi-scale triangle representation, which contains the curvature features of the contour, the texture features and the shape area features of the leaf image.
2) Our method provides a multi-scale descriptor and an outstanding ability to describe the overall layout and local details of the leaf image, which conforms to the multi-scale property of human to recognize objects.
3) Only a few pixels of the leaf image to participate in the calculation, so our method has very low computational cost and hardware consumption.
4) An adaptive KNN for optimization is applied to greatly improve the retrieval rate of the flavia leaf dataset, which combined the KNN classifier with the optimization process.

The rest of this article is arranged as follows: we review the related work of leaf recognition in section II. The preprocessing work is shown in section III and the multi-scale triangle representations are elaborated in section IV. In section V a new adaptive KNN for optimization method is presented. The performance of our method and our method with optimization are experimented in section VI. Finally, the paper ends with concluding remarks in section VII.

II. RELATED WORKS

In this section, we review the current popular leaf recognition methods. Generally speaking, leaf features contain leaf contour features, leaf shape area features, and texture features.

There are many types of leaf contour features. On the Swedish leaf dataset, Mouine *et al.* [1] used the triangles on the contour points to calculate the side length, angle, area and other parameters to describe the contour of the shape, and achieved 96.53% accuracy; Hu *et al.* [4] proposed a multi-scale distance matrix description method, in which the multi-scale distance matrix is calculated on the contour points to represent the contour spatial distribution, and they reported 93.60% accuracy; Wang *et al.* [6] extracted multiscale arch height(MARCH) from each contour point to provide a contour curvature description, and attained 97.33% accuracy; Zhao *et al.* [7] combined multiscale Gaussian convolution on the contour points and the shape mode dictionary which proved to be better than matching extracted features and an accuracy of 97.10% is achieved; Ling *et al.* [13] attempted to apply inner distance to measure shapes similarity instead of Euclidean distance, and obtained an accuracy of 94.13%; Yu *et al.* [5] attained 95.67% accuracy by using combined feature of contour and venation for leaf image identification; Md.Ajij *et al.* [8] analyzed the shape similarity by using the Pearson Correlation Coefficient calculated from leaf boundary pixels; Compared with the original triangle representation, Yang *et al.* [11] proposed a symbol matrix and a triangle center distance matrix to represent the leaf contour features, and obtained an accuracy of 97.27%; The results of the above-mentioned method are not satisfactory. Using only leaf contour feature information, lack of features such as shape area features and leaf texture features, cannot fully and effectively express leaf features.

In terms of the feature of the leaf shape area, Liu *et al.* [14] converted the binary function of the leaf binary image into a Fourier power spectrum to describe the leaf shape area features, and obtained 92.27% accuracy on the public Swedish leaf dataset; Kumar *et al.* [14] calculated the aspect ratio, area concave-convex ratio, and minor axis length based on the contour of the leaf, and achieved 95.42% accuracy on the Flavia leaf dataset by using the AdaBoost classifier. Chaki and Parekh [42] combined Moment Invariants and Centroid-Radii model for a classification accuracy of 97.9%. ArunPriya *et al.* [32] presented a hybird method of representing the basic geometric features, morphological features, vein features of leaf with SVM classifier and achieved 94.5% accuracy on the flavia leaf dataset. Neto *et al.* [33] used Elliptic Fourier to identify young soybean, sunflower, redroot pigweed and velvetleaf plants and obtained 89.2% accuracy. Aptoula and Yanikoglu [34] presented morphological covariance on the leaf contour profile and circular covariance histogram and achieved 56.09% on ImageCLEF2012 datset. Du. *et al.* [35] extracted Digital morphology features to classify 20 species of plant leaves and the result was good. The above method based on the leaf shape area features has a single description ability and a poor recognition effect.

There are many ways to extract leaf texture features, including Local binary pattern (LBP) [31], Gray level co-occurrence matrix (GLCM) [27]–[30] and Gabor features, etc. In [31], Modified Local Binary Patterns (MLBP) were put forward for the extraction of texture feature from plant leaves. Chaki *et al.* [9] combined the Gabor filters and the gray-level co-occurrence matrix to represent the texture features of leaves, and obtained 97.6% accuracy rate on the Flavia leaf dataset with MLP and NFC classifiers. Prasvita and Herdiyeni [37] applied Fuzzy Local Binary Pattern (FLBP) and the Fuzzy Color Histogram (FCH) in order to identify medicinal plants and achieved 74.51% accuracy. Man *et al.* [38] extracted GLCM from the image after digital wavelet transform and obtained 92.2% accuracy on 24 species of plant leaves. Tang *et al.* [20] extracted the GLCM and LBP of green tea leaves. The leaf image is often compressed by the above methods due to the time complexity. Leaf texture features are easily affected by external disturbances so that it is difficult to use it as a means to identify the leaves alone.

Many researchers tried to extract multiple features to recognize leaves. El Massi *et al.* [29] combined color feature, shape feature and texture feature and obtained 91.46% accuracy Leaf image of diseases and insect pests. Lee *et al.* [39] combined leaf contour centroid distance, Geometric and Digital and Morphological Features and achieved 95.44% accuracy on the Flavia leaf dataset. Polar Fourier transform, color moments, vein features are applied in leaf recognition and the authors [40] reported 93.13% accuracy on the Flavia dataset. Shape features, FD and multiscale distance matrix was shown in [41], and the authors attained 94.62% accuracy on the Flavia dataset. Aakif *et al.* [2] extracted various features such as the shape, texture, and color features of the leaves, the optimal feature vector is calculated by the ant colony algorithm, and 96.25% accuracy rate is attained on the Flavia leaf dataset. Turkoglu *et al.* [12] combined the Fourier descriptors, color features, and gray-level co-occurrence matrix features of the leaves to achieve 99.10% accuracy on the Flavia leaf dataset by using the ELM classifier. The above-mentioned methods based on multiple features get higher accuracy, but they have the problem of high time complexity and redundancy.

In short, only using one of contour feature, texture feature or shape feature, the recognition effect is not satisfactory. Although some methods consider extracting multiple characteristics of leaves, the texture features and shape features can only describe the overall layout of the leaves and lack detail descriptions. Moreover, these algorithms rely on the optimizers or neural networks to improve recognition accuracy. There are still many problems including that the time complexity is high, training does not converge or falls into local optimization. In order to solve the above problems, a simple, effective, efficient and robust leaf recognition algorithm is proposed in this paper.

III. IMAGE PREPROCESSING

Before extracting features from leaf images, preprocessing is first required. In this section we briefly introduce the preprocessing of a leaf image, the steps are as follows:

1) Gray-level image.
   To convert the RGB image into a gray-level image, we obtain the grayscale matrix representation by the following formula:

$$G = R * 0.3 + G * 0.59 + B * 0.11 \tag{1}$$

   where R, G, B are integers between 0 and 255 and correspond to the color of the pixel, 0.3, 0.59, 0.11 are weight parameters.
2) Convert it into a binary matrix representation.
   By using the threshold segmentation function, a binary matrix representation is obtained:

$$B(x_i, y_i) = \begin{cases} 1 & G(x_i, y_i) < T \\ 0 & G(x_i, y_i) \geq T \end{cases} \tag{2}$$

   where T is the threshold, (x_i, y_i) is the coordinates of the pixels.
3) Generate its leaf contour points.
   By applying the classic outer contour tracking function, N contour points are uniformly sampled counter-clockwise, and the contour points p is obtained: $p = \{(x_i, y_i)|0 \leq i \leq N-1\}$

The flow chart of leaf image preprocessing is shown in Fig.2.

IV. MULTISCALE TRIANGLE REPRESENTATION

The multi-scale triangle representation mainly extracts the contour curvature features, texture features and shape area feature of each leaf image. In this section we elaborate on the contour curvature features TUA and TVA, texture features TGA and TGSD, and shape area feature TSLI (see in section IV.*A*-IV.*D*).

The boundary of each shape is uniformly sampled into contour points $p_0, p_1, \ldots, p_{N-1}$ (see Fig. 3(a)), each contour point p_i is related to K triangles of different sizes(see

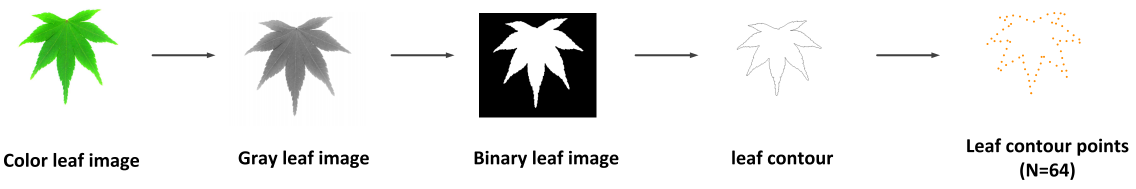

FIGURE 2. Leaf Image preprocessing.

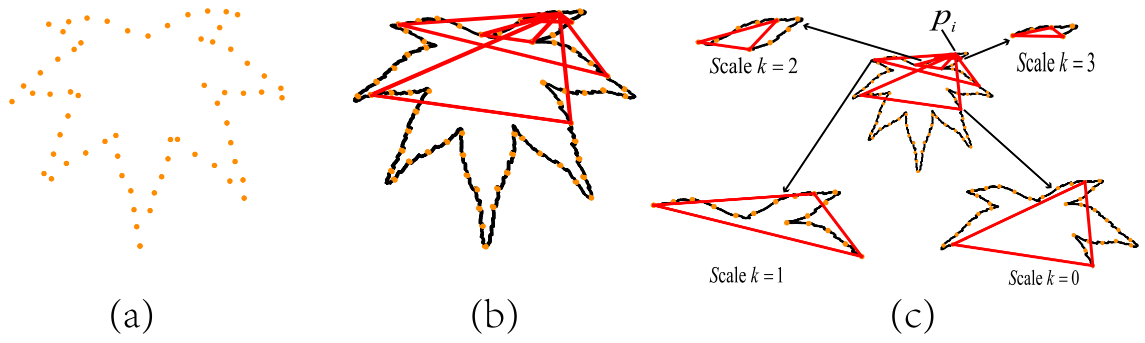

FIGURE 3. Multiscale triangular representation. (a) Uniformly sampled leaf contour with $N = 64$ points. (b) p_i is represented by $K = 4$ triangles. (c) Four triangles with scale value $x(0) = 16, x(1) = 8, x(2) = 4, x(3) = 2$.

Fig. 3(b)).The function $x(k)$ is defined as follows:

$$x(k) = \left\lceil \frac{N/4}{2^k} \right\rceil, \quad 0 \le k \le K-1 \tag{3}$$

where k is the valuable of scale, $x(k)$ is the scale value which represents the number of contour points between contour point p_i and contour point $p_{i+x(k)}$ or contour point $p_{i-x(k)}$ in the triangle. The smaller the scale value $x(k)$, the smaller the triangle, which can capture the local feature of the leaf image; conversely, the larger the scale value $x(k)$, the larger the triangle, which can capture the overall feature of the leaf image(see Fig. 3(c)). $T_i{}^k$ is defined as a triangle composed of contour point p_i, $p_{i+x(k)}$ and $p_{i-x(k)}$. Its scale value is $x(k)$ and its side lengths are $a_i^k = |\overline{p_i p_{i-k}}|$, $b_i^k = |\overline{p_i p_{i+k}}|$, $c_i^k = |\overline{p_{i-k} p_{i+k}}|$ (see Fig.4). T^k is a set of triangles with contour points $p_0, p_1, \dots, p_{N-1}$ as the vertex and their scale value are $x(k)$.

Each sample is represented by five triangle representations: Γ = (TUA, TVA, TGA, TGSD, TSLI) where TUA is represented by K representations: TUA = (TUA(T^0), ..., TUA(T^{K-1})) where TUA(T^k) is represented by N variables: TUA(T^k) = (TUA(T_0^k), ..., TUA(T_{N-1}^k)).

Therefore TUA = (TUA(T_0^0), ..., TUA(T_{N-1}^0), ..., TUA(T_0^{K-1}), ..., TUA(T_{N-1}^{K-1})), the remaining TVA, TGA, TGSD and TSLI are also defined in the same way.

A. TRIANGLE UNSIGNED AREA REPRESENTATION (TUA)

For each triangle, TUA(T) = AREA(T), TAR is the signed area of the triangle in [1], but due to the sampling deviation of the contour points, TUA represents the unsigned triangle area here, and its description ability is better than TAR [1].

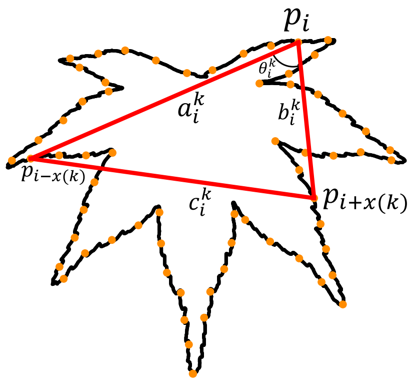

FIGURE 4. $T_i{}^k$ representation.

$$TUA(T_i^k) = \frac{1}{2} ABS\left(\begin{vmatrix} x_{i-x(k)} & y_{i-x(k)} & 1 \\ x_i & y_i & 1 \\ x_{i+x(k)} & y_{i+x(k)} & 1 \end{vmatrix} \right) \tag{4}$$

Because TUA(T_i^k) changes with the size of the triangle T_i^k, we should normalize it as follows:

$$TUA(T_i^k) = \frac{TUA(T_i{}^k)}{AVG(\sum_{i=0}^{N-1} a_i^k \times b_i^k)} \tag{5}$$

where $a^k = \{a_i^k | 0 \le i \le N-1, a_i^k \in R\}$, $b^k = \{b_i^k | 0 \le i \le N-1, b_i^k \in R\}$ are the side length set.

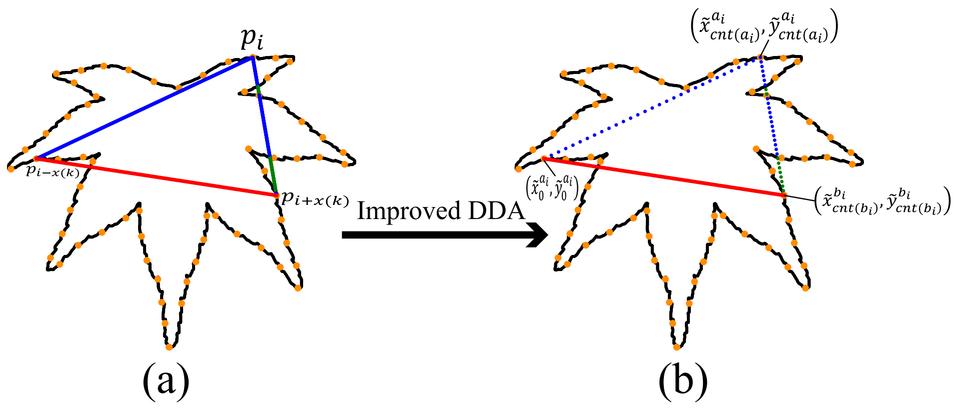
p_i
$p_{i-x(k)}$
$p_{i+x(k)}$
Improved DDA
$(\tilde{x}^{a_i}_{cnt(a_i)}, \tilde{y}^{a_i}_{cnt(a_i)})$
$(\tilde{x}^{a_i}_0, \tilde{y}^{a_i}_0)$
$(\tilde{x}^{b_i}_{cnt(b_i)}, \tilde{y}^{b_i}_{cnt(b_i)})$
(a)
(b)

FIGURE 5. TGSSL representation. (a) Grayscale distribution on two sides a_i^k and b_i^k of the triangle. (b) Pixels sampled by the improved DDA algorithm on both sides a_i^k and b_i^k of the triangle with Scale $H = 20$. (The blue pixels fall within the shape area, and the green pixels fall outside the shape area).

B. TRIANGLE REPRESENTED BY A VERTEX ANGLE (TVA)

TVA representation is the vertex angle of the triangle $T_i{}^k$ (see Fig. 4), and it also describes the curvature features of leaf contour. Let the vertex angle between a_i^k and b_i^k is θ_i^k and its calculation formula is as follows:

$$\theta_i^k = \arccos(\frac{(a_i^k)^2 + (b_i^k)^2 - (c_i^k)^2}{2 \times a_i^k \times b_i^k}) \tag{6}$$

In order to improve the fault tolerance of TVA, the calculation formula is:

$$TVA(T_i{}^k) = \widetilde{\theta}_i^k = \arccos((\frac{(a_i^k)^2 + (b_i^k)^2 - (c_i^k)^2}{2 \times a_i^k \times b_i^k})^2) \tag{7}$$

C. TRIANGLE REPRESENTED BY THE GRAYSCALE STATISTICS OF TWO SIDE LENGTHS (TGSSL)

TGSSL representation describes the texture feature of the leaf, including two new representations which are represented by the gray average on triangle(TGA) and the gray standard deviation on triangle(TGSD).The above two representations together describe the gray distribution on the two sides a_i^k and b_i^k of the triangle(see Fig. 5(a)).In order to obtain the gray distribution on the triangle, the Digital Differential Analyzer (DDA) algorithm is applied to sample the coordinate points on the two sides a_i^k and b_i^k. The coordinate points on the triangle set T^k are sampled in parallel, and the DDA algorithm makes the following improvements:

1) Calculate $cnt(a_i^k) = \lfloor AVG(a_0^k, a_1^k, \ldots, a_{N-1}^k) \rfloor$, $cnt(b_i^k) = \lfloor AVG(b_0^k, b_1^k, \ldots, b_{N-1}^k) \rfloor$, and we define a scale H to reduce the number of sampling points:

$$cnt(a_i^k) = cnt(a_i^k)/H, \quad cnt(b_i^k) = cnt(b_i^k)/H \tag{8}$$

2) Calculate the increment on the x-axis:

$$dx(a_i^k) = \frac{x_i - x_{i-k}}{cnt(a_i^k)}, \quad dx(b_i^k) = \frac{x_{i+k} - x_i}{cnt(b_i^k)} \tag{9}$$

Calculate the increment on the y-axis:

$$dy(a_i^k) = \frac{y_i - y_{i-k}}{cnt(a_i^k)}, \quad dy(b_i^k) = \frac{y_{i+k} - y_i}{cnt(b_i^k)} \tag{10}$$

3) Calculate the sampling coordinates $(\widetilde{x}_q^{a_i}, \widetilde{y}_q^{a_i})$ on one side a_i^k of the triangle:

$$\begin{aligned} \widetilde{x}_q^{a_i} &= \lfloor x_{i-k} + q \times dx(a_i^k) \rfloor, \quad 0 \le q \le cnt(a_i^k) \\ \widetilde{y}_q^{a_i} &= \lfloor y_{i-k} + q \times dy(a_i^k) \rfloor, \quad 0 \le q \le cnt(a_i^k) \end{aligned} \tag{11}$$

Sampling coordinates $(\widetilde{x}_r^{b_i}, \widetilde{y}_r^{b_i})$ on the other side b_i^k:

$$\begin{aligned} \widetilde{x}_r^{b_i} &= \lfloor x_i + r \times dx(b_i^k) \rfloor, \quad 1 \le r \le cnt(b_i^k) \\ \widetilde{y}_r^{b_i} &= \lfloor y_i + r \times dy(b_i^k) \rfloor, \quad 1 \le r \le cnt(b_i^k) \end{aligned} \tag{12}$$

More importantly, there is a contour point (x_i, y_i) on side a_i^k, but not on side b_i^k. Therefore, the number of points sampled on side a_i^k is $cnt(a_i^k)$+1, and the number of points sampled on side b_i^k is $cnt(b_i^k)$. The set of coordinate points $\Re$ sampled on the triangle $T_i{}^k$ is: $\Re = \{(\widetilde{x}_0^{a_i}, \widetilde{y}_0^{a_i}), \ldots, (\widetilde{x}^{a_i}_{cnt(a_i^k)}, \widetilde{y}^{a_i}_{cnt(a_i^k)}), (\widetilde{x}_1^{b_i}, \widetilde{y}_1^{b_i}), \ldots, (\widetilde{x}^{b_i}_{cnt(b_i^k)}, \widetilde{y}^{b_i}_{cnt(b_i^k)})\}$ (see Fig. 5(b)).

After simplification, the set is expressed as: $\Re = \{(\widetilde{x}_t, \widetilde{y}_t)|0 \le t \le cnt(a_i^k) + cnt(b_i^k)\}$

It is obvious that the time cost of mapping the set $\Re$ to the gray value subset $\widetilde{G}$ and the binary subset $\widetilde{B}$ is very low. It only needs the grayscale matrix representation G(given in formula (1)), the binary matrix representation B(given in formula (2)) and the set $\Re$ to obtain the gray value subset $\widetilde{G} = \{G[\widetilde{\Re}_t]\}|0 \le t \le cnt(a_i^k) + cnt(b_i^k)\}$ and the binary subset $\widetilde{B} = \{B[\widetilde{\Re}_t]|0 \le t \le cnt(a_i^k) + cnt(b_i^k)\}$. When $\widetilde{B}_t = 0$, the coordinate point $(\widetilde{x}_t, \widetilde{y}_t)$ is outside the leaf image, and when $\widetilde{B}_t = 1$, the coordinate point $(\widetilde{x}_t, \widetilde{y}_t)$ is inside the leaf image.

However, there are noises in the binary subsets $\widetilde{B}$ corresponding to the triangle set T^k, and they are blur filtered to

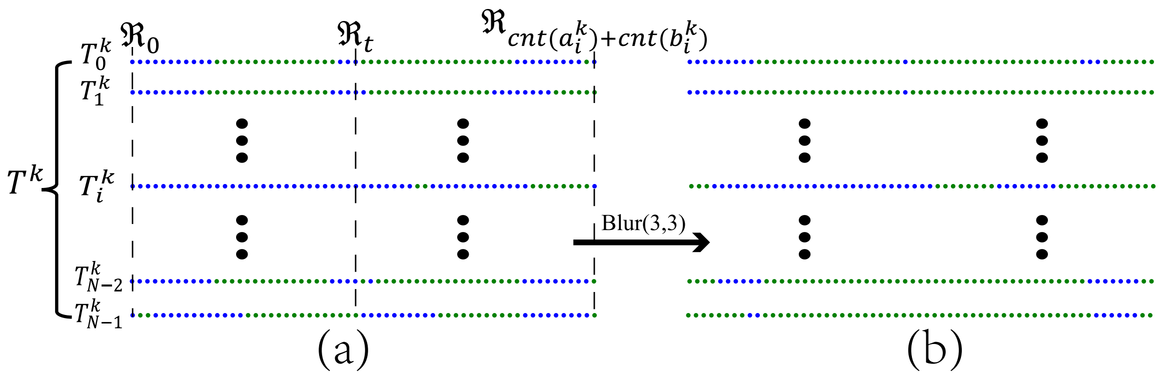

FIGURE 6. The binary subset $\widetilde{B}$ corresponding to each triangle $T_i{}^k$ in the triangle set T^k. (a) Before blur filtering. (b) After blur filtering. Pixels are blue when $\widetilde{B}_t = 1$, pixels are green when $\widetilde{B}_t = 0$ in triangle T^k.

eliminate noise, here the window size is 3 × 3. The formula for rounding after filtering is as follows(see Fig. 6(b)):

$$\widetilde{B}_t = \begin{cases} 0 & 0 < \widetilde{B}_t < 1 \\ 1 & otherwise. \end{cases}$$

According to the binary subset $\widetilde{B}$ and the gray value subset $\widetilde{G}$, TGA($T_i{}^k$) and TGSD($T_i{}^k$) are calculated as follows ($cnt = cnt(a_i^k) + cnt(b_i^k)$):

$$TGA(T_i{}^k) = \frac{\sum_{t=0}^{cnt} G_t \times \widetilde{B}_t}{\sum_{t=0}^{cnt} \widetilde{B}_t} \tag{13}$$

$$TGSD(T_i{}^k) = \sqrt{\frac{\sum_{t=0}^{cnt} ((G_t - TGA(T_i^k)) \times \widetilde{B}_t)^2}{\sum_{t=0}^{cnt} \widetilde{B}_t}} \tag{14}$$

It can be seen that TGA representation represent the average level of gray distribution, and TGSD representation describes the fluctuation of the gray distribution on the triangle.

D. TRIANGLE REPRESENTED BY SIDE LENGTH INTEGRAL (TSLI)

TSLI representation is the distribution of pixels on the two sides a_i^k and b_i^k of the triangle $T_i{}^k$, which describes the shape area feature of the leaf image. According to the binary subset $\widetilde{B} = \{B[\Re_t]|0 \le t \le cnt(a_i^k) + cnt(b_i^k)\}$ in section IV.*C*, TSLI($T_i{}^k$) is calculated as follows:

$$TSLI(T_i{}^k) = \frac{\sum_{t=0}^{cnt(a_i^k)+cnt(b_i^k)} \widetilde{B}_t}{cnt(a_i^k) + cnt(b_i^k) + 1} \tag{15}$$

The result of TSLI($T_i{}^k$) is actually the proportion of blue pixels in the row(see Fig. 6(b)). All five representations are RST invariant(shown in VI.*E*).

E. FEATURE NORMALIZATION

We introduced five triangle representations in section IV.*A*-IV.*D*. But the results of TUA(T^k), TVA(T^k), TGA(T^k), TGSD(T^k) and TSLI(T^k) are all related to the starting position of the contour point. Perform Fast Fourier Transform(FFT) on the above five representations to obtain a Fourier coefficient sequence $\underline{\text{TUA}}(T^k)$, $\underline{\text{TVA}}(T^k)$, $\underline{\text{TGA}}(T^k)$, $\underline{\text{TGSD}}(T^k)$ and $\underline{\text{TSLI}}(T^k)$ with length N and they're rotation invariant, because the change of the starting point of the contour does not change the Fourier coefficient. In order to reduce the influence of noise, we only take the first $M (M \ll N)$ coefficients, and the representations are as follows:

$$\begin{aligned} \underline{\text{TUA}}(T^k) &= (\underline{\text{TUA}}(T_0^k), \ldots, \underline{\text{TUA}}(T_{M-1}^k)) \\ \underline{\text{TVA}}(T^k) &= (\underline{\text{TVA}}(T_0^k), \ldots, \underline{\text{TVA}}(T_{M-1}^k)) \\ \underline{\text{TGA}}(T^k) &= (\underline{\text{TGA}}(T_0^k), \ldots, \underline{\text{TGA}}(T_{M-1}^k)) \\ \underline{\text{TGSD}}(T^k) &= (\underline{\text{TGSD}}(T_0^k), \ldots, \underline{\text{TGSD}}(T_{M-1}^k)) \\ \underline{\text{TSLI}}(T^k) &= (\underline{\text{TSLI}}(T_0^k), \ldots, \underline{\text{TSLI}}(T_{M-1}^k)) \end{aligned}$$

The calculation formula of TUA representations is updated as follows:

$$\underline{\text{TUA}} = (\underline{\text{TUA}}(T^0), \ldots, \underline{\text{TUA}}(T^{K-1}))$$

The remaining TVA, TGA, TGSD and TSLI representations are calculated in the same way. Finally, each leaf image is represented by a triangular representation Γ with a length of 5 × K × M: $\Gamma = (\underline{\text{TUA}}(T^0), \ldots, \underline{\text{TUA}}(T^{K-1}), \ldots, \underline{\text{TSLI}}(T^0), \ldots, \underline{\text{TSLI}}(T^{K-1}))$.

Because TUA, TVA, TGA, TGSD and TSLI have different orders of magnitude, it is necessary to normalize the representation Γ to prevent the weight of a certain triangle representation from becoming too large. The steps for normalizing the TUA representation are as follows:

1) Calculate minimum and maximum values of TUA of the training set: $V\min_j = \min(TUA_j)$, $V\max_j = \max(TUA_j)$, $0 \le j \le s - 1$, j is the index number of the training set, s is the size of training set.
2) Calculate $V\min$ and $V\max$:

$$V\min = (Vmin_0, V\min_1, \ldots, V\min_{s-1})$$
$$V\max = (V\max_0, V\max_1, \ldots, V\max_{s-1})$$

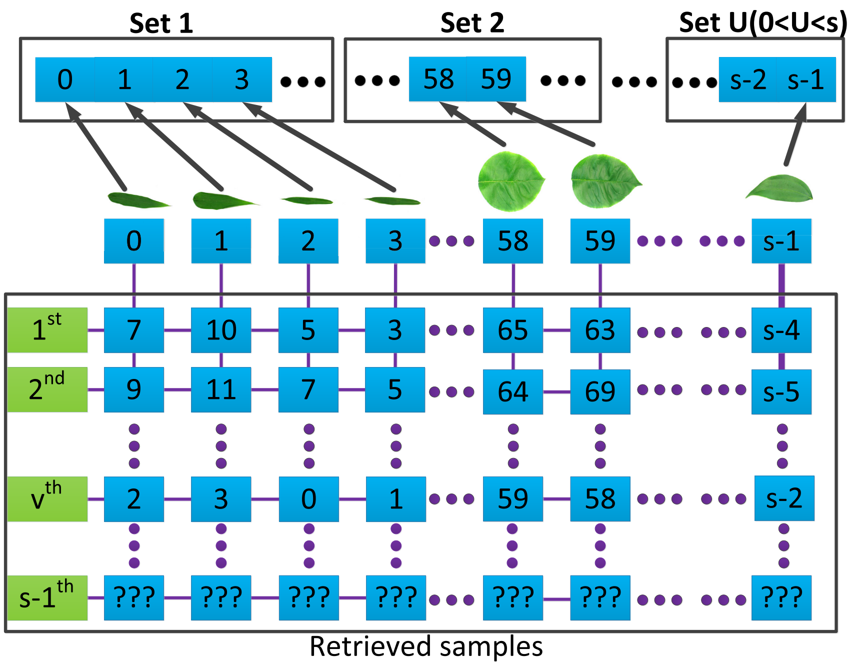
Set 1
Set 2
Set U(0<U<s)
0 1 2 3 •••
••• 58 59 •••
••• ••• s-2 s-1
0 1 2 3 ••• 58 59 ••• ••• s-1
1st 7 10 5 3 ••• 65 63 ••• ••• s-4
2nd 9 11 7 5 ••• 64 69 ••• ••• s-5
vth 2 3 0 1 ••• 59 58 ••• ••• s-2
s-1th ??? ??? ??? ??? ••• ??? ??? ••• ••• ???
Retrieved samples

FIGURE 7. Schematic diagram of dividing the dataset. s is the number of samples in the dataset, $1 \le v \le s-1$.

3) Replace TUA with $\frac{TUA - V\min}{V\max - V\min}$, including training set and test set.

The normalization of the remaining TVA, TGA, TGSD and TSLI is the same, and finally each representation is given a different weight as follows: $\Gamma = (2 \times \text{TUA}, 2 \times \text{TVA}, \text{TGA}, \text{TGSD}, \text{TSLI})$, the weight setting is related to the experiment in Section VI.

F. MATCHING METHOD

Any two leaf images are represented as $\Gamma_\alpha = (TUA_\alpha, TVA_\alpha, TGA_\alpha, TGSD_\alpha, TSLI_\alpha)$ and $\Gamma_\beta = (TUA_\beta, TVA_\beta, TGA_\beta, TGSD_\beta, TSLI_\beta)$, α, β are the index number in the dataset, their distance metric is based on Bray-curtis Distance:

$$dis(\alpha, \beta) = \frac{|\Gamma_\alpha - \Gamma_\beta|}{\Gamma_\alpha + \Gamma_\beta} \tag{16}$$

where |.| represents the L1 distance, the smaller the result of this function, the more similar the two leaf images.

V. KNN FOR OPTIMIZATION

The retrieval rate of dataset is often related to the distance between different samples. In order to improve the retrieval rate, we propose a new adaptive KNN for optimization. This method uses KNN classifier to divide the dataset, and then optimize the samples representation. The process of KNN classifier to divide the dataset is shown in Fig.7. Each sample and v^{th} match are classified into the same sets, and finally the dataset is divided into U sets. $1 \le v \le s-1, 0 < U < s$.

KNN for optimization can be described by Algorithm 1 in detail, which consists of two steps. The first step is the division of the dataset, and the second step is the optimization of the samples representation matrix Γ.

It is important to note that the v^{th} match of the method changes dynamically based on the number of samples per class in the dataset. Therefore, we define the formula as follows:

$$v = \left\lfloor \chi \frac{s}{C} \right\rfloor \tag{17}$$

where C is the number of classes in the dataset, and s is the number of samples in the dataset, χ is the ratio variable.

VI. EXPERIMENTAL RESULT

In section VI.A, we set the parameters of our method. In section VI.B Our method and Our method with optimization (using KNN for optimization in section V) are compared with the state-of-the-art methods on five datasets, including Swedish [22], Flavia [23], MPEG-7 [24], Kimia99 [25] and Kimia216 [25]. In Section VI.C, our method is proved to be robust by adding the salt-and-pepper noises. In Section VI.D, our method and our method with optimization are compared with others on computational cost. At the end, our method is proved to be invariant to scaling, translation and rotation.

A. SETTINGS

Our method is implemented under python3.0 and runs on i5-6300HQ CPU with 2.3GHz and 8GB memory. The parameters are set as: the number of contour points $N = 256$, the number of scales $K = 7$, the scaling factor of the number of triangle T_i^k acquisition points $H = 5$, take the first $M = 7$ Fourier coefficient of the Fourier transform sequence.

Algorithm 1: Generation Process of KNN for Optimization

```
Input: the samples representation matrix $\Gamma$
       ($row : s, col : 5 \times M \times K$)
Output: the samples representation matrix after
       optimization $\widetilde{\Gamma}$ ($row : s, col : C - 1$)
begin
    ·Divide the dataset according to the samples
     representation matrix $\Gamma$
    ·Initialization
    $Parent\,[0, s-1] := 0, 1, \ldots, s-1$,where $s$ is the size of dataset.
    ·Using KNN classifier to divide data set.
    for $index\, j \in [0, s-1]$ do
        $Union(j$,Similar set $\phi_j[v])$,where $\phi_j$ is a similar
        set retrieved by sample $\Gamma_j$ with KNN classifier
        of $s-1$ length, $v$ is an index of similar set $\phi_j$,
        $0 \le v \le s-1$;
    for $index\, j \in [0, s-1]$ do
        $Find(j)$;
    ·Combine the samples representation matrix $\Gamma$ and
     $Parent$ to perform optimization.
    $mean_i = \frac{1}{U_i}\sum_{u=0}^{U_i-1}\Gamma_{iu}$, where $U_i$ is the number of the
     $i^{th}$ category in the divided dataset(see Fig.7), $\Gamma_{iu}$ is
     the $u^{th}$ sample of the $i^{th}$ category, $0 \le i \le U-1$.
    $mean = \frac{1}{s}\sum_{j=0}^{s}\Gamma_j$
    ·Calculate between-class scatter matrix:
     $S_b = \sum_{i=0}^{U-1} U_i\,(mean_i - mean)\,(mean_i - mean)^T$,
     where $U$ is the number of categories in the divided
     dataset in Fig.7.
    ·Calculate within-class scatter matrix
     $S_w = \sum_{i=0}^{U-1}\sum_{u=0}^{U_i-1}(mean_i - \Gamma_{iu})(mean_i - \Gamma_{iu})^T$
    ·Calculate feature values and feature vectors
    $S = S_w^{-1}S_b$
    ·$\lambda_1, \lambda_2, \cdots, \lambda_{C-1}$ and $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{C-1}$ are feature
     values and feature vectors of $S$ sorted by $\lambda$ (largest
     to smallest), $C$ is the number of categories in the
     dataset.
    ·Redistribute the weight of matrix coefficients,
     remove redundant parts, and calculate the optimal
     matrix $\widetilde{\Gamma}$
    $\widetilde{\Gamma} = \Gamma \times [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{C-1}]^T$
Fuctions:
• function $Find$(x)                • function $Union$(x, y)
if $Parent$(x) != x:                    xRoot := $Find$(x)
    $Parent$(x) = $Find$($Parent$[x])       yRoot := $Find$(y)
    return $Parent$[x]                  $Parent$[xRoot] := yRoot
```

Different data sets have different v^{th} match in Our method with optimization, in addition to the Swedish dataset, the values are shown in Table 1.

TABLE 1. Different v^{th} in our method with optimization in different dataset.

Dataset	Number of samples	Number of categories	v
Flavia	1907	32	8
MPEG-7	1400	70	2
Kimia99	99	9	1
Kimia216	216	18	1

TABLE 2. Accuracy of different approaches on the Swedish leaf dataset.

Method	Accuracy (%)
MDM [4]	93.60
IDSC [13]	94.13
MCR [5]	95.67
TSLA [1]	96.53
HATISI&RS [19]	96.53
GFPC [8]	97.00
Pattern Counting [7]	97.10
TDR [11]	97.27
MARCH [6]	97.33
GMM+soft Clustering+Ts1 [14]	98.33
CSR [16]	99.50
Our Method	**99.35**

B. EXPERIMENTS

1) THE SWEDISH LEAF DATASET

This public dataset has a total of 1125 leaves, including 15 classes, and each class has 75 leaves (see Fig. 8(a)). It is the same as [1], [5]–[8], [12], [15], [16], [18], each class on the Swedish is randomly divided into training samples and test samples at a ratio of 1:2, so there are 375 samples in the training set, and 750 samples in the test set. The nearest neighbor (1-NN) classifier is used for classification, that is, in the returned most similar samples, if the first sample is in the same class as the classificated sample, the sample is correctly classified. Since the division method is random selection, there is a lot of contingency. Therefore, this experiment will be repeated 100 times, and the average result of 100 experiments is used as the final result.

Table 2 shows the accuracy of different methods on the Swedish dataset. The accuracy of our method reaches 99.35%. Compared with the CSR, our method misjudges only one more leaf image on average. But the computational cost of ours is quite low, requiring only 153.97ms to identify each image, and while CSR requires 2130ms (See VI.*D*), which is ten times of ours. Therefore, this experiment proves that our method has high accuracy and is very suitable for real-time tasks.

2) THE FLAVIA LEAF DATASET

This public dataset has 32 classes, each class has about 50-77 leaf images, and a total of 1907 leaf images. The leaf images were photographed in the Yangtze River Delta region of China. The leaf categories include Phyllostachys pubescens, Luan Shu, Holly Daguo, Pittosporum vulgare and Wintergreen (see Fig. 8(b)). We conduct two experiments, one is retrieval experiment and the other is classification experiment.

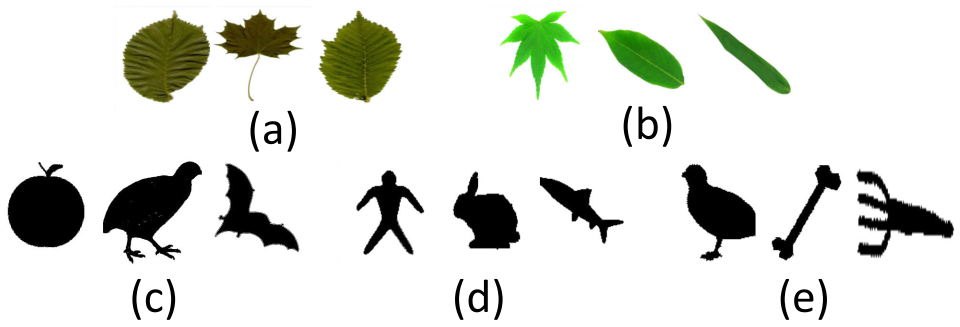

FIGURE 8. Examples of different leaf Dataset. (a) From the Swedish Leaf Dataset. (b) From the Flavia Leaf dataset. (c) From the MPEG-7 dataset (d) From the kimia99 dataset (e) From the kimia216 dataset.

TABLE 3. MAP values of different methods on the Flavia leaf dataset.

Method	Map (%)
MDM [4]	59.10
IDSC [13]	59.90
TDR [11]	69.61
TSLA [1]	69.90
MARCH [6]	73.00
R-angle Description [15]	74.70
MCR [5]	81.48
CSR [16]	94.11
Our Method	**84.76**
Our Method with Optimization	**94.48**

a: RETRIEVAL EXPERIMENT

Like other methods in Table 3, MAP based on leave-one-out method is used for the evaluation of experimental. Table 3 shows the comparison between our method and the state-of-the-art methods on the Flavia. Except for the CSR, our method attains the highest MAP value of 84.76%. While our method with optimization attains 94.48% of MAP value which is better than the CSR. CSR is extremely time-consuming. It takes 2.13s to recognize each picture, while our method with optimization only takes 110.80ms (see in Section VI.*D*) to recognize. Therefore, our method attains a high retrieval rate, which is very suitable for real-time tasks and large-scale image retrieval tasks. Also, KNN for optimization can be proved to greatly improve the retrieval rate of the Flavia dataset.

The multi-scale triangle representation Γ proposed in this paper include TUA, TVA, TGA, TGSD and TSLI representations. To find out the contribution of each representation, we do the following experiments. Table 4 shows the comparison of the MAP values of each representation on the Flavia. TUA obtains the highest MAP value, and TGA obtains the worst result. The MAP value of TVA and TUA is above 70% and others below 70%, take 70% MAP value as base level, the weights of TVA and TUA are set to 2, and the weights of others are 1.

The KNN for optimization is related to "v^{th} match", so that we do the following experiments. Fig. 10(a) shows the European distribution of 10 similar samples.

TABLE 4. MAP of various feature on the Flavia leaf dataset (MAP* is a version of our method with optimization).

Representation	Map (%)	MAP* (%)
TGA	35.34	41.97
TGSD	39.35	43.67
TSLI	55.07	60.30
TVA	74.77	77.32
TUA	75.90	78.98

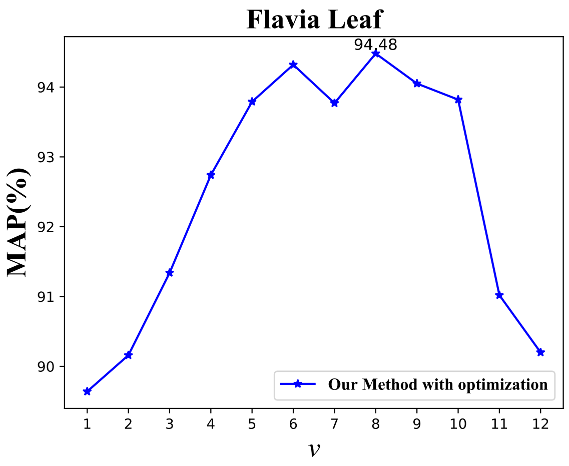

FIGURE 9. Influence of "v^{th} match" on MAP on the Flavia leaf dataset.

v varies from 1 to 12. when $v = 1$, the distance between the similar sample and the retrieved sample(red star) is reduced compared to Fig. 10(a), but the MAP value is still relatively low (see Fig. 9). As v grows, the distance between the retrieved sample and the similar samples (for example, the black hexagon) gradually decreases (see Fig. 10(c)). When $v = 8$, the MAP value reaches the maximum value of 94.48% (see Fig. 9). When $v > 8$, the retrieved sample and the similar samples gradually increases (see Fig. 10(d)), and the MAP value begin to decrease (see Fig. 9). When v is equal to the number of samples in the dataset, the distribution of the 10 samples become chaotic and random, and the MAP value will tend to 0.

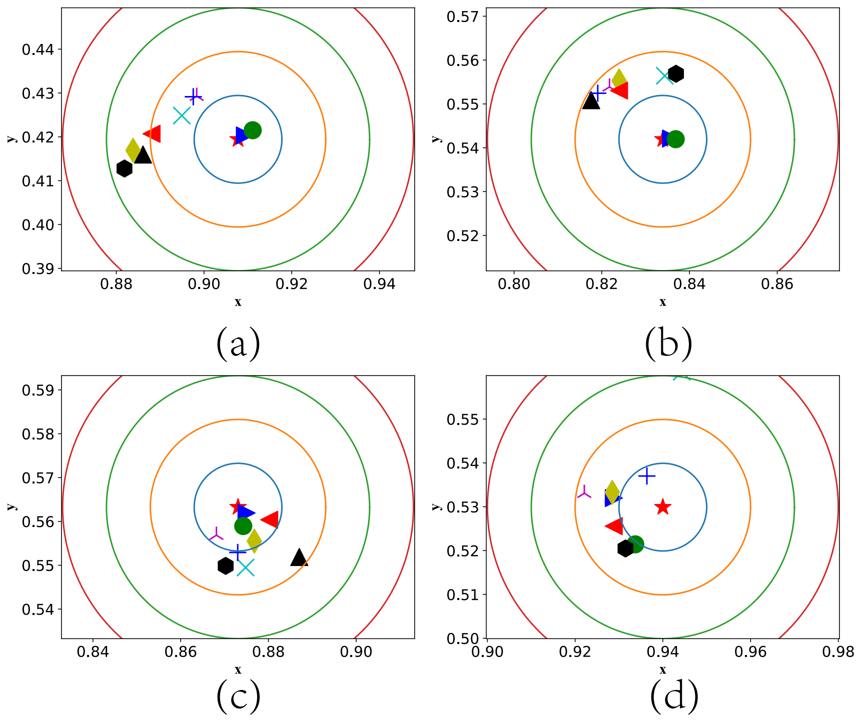

FIGURE 10. The impact of different "v^{th} match" on 10 similar samples in the same class. (a) Our method. (b) Our method with optimization ($v = 1$). (c) Our method with optimization ($v = 8$). (d) Our method with optimization ($v = 19$) (Here the center of the circle is the retrieved sample and the interval of each circle is 0.02 in L2 Distance.)

It can be seen from experiments that v^{th} match is a balance point which is related to the number of samples in the each class because it requires $v^{th}(\overleftarrow{v} < v)$ match in the same class as v^{th} match, and $v^{th}(\overrightarrow{v} > v)$ match in a different class with v^{th} match. Therefore, the scaling factor χ in formula (17) can be calculated as: $\chi = \frac{8}{s/C}$, where C is the number of classes in the dataset, and s is the number of samples in the dataset. Taking the Flavia dataset as a benchmark to get $\chi = 0.13$. (see in Table 1).

b: CLASSIFICATION EXPERIMENT

In this part, we divide the training set and test set on the Flavia as experimental evaluation. Because many methods have different size of training set and test set, in order to have a better comparison, we apply different division ratios according to these methods. Table 5 shows the comparison between our method and the state-of-the-art methods in the accuracy on the Flavia. No matter which division ratio, our method has the highest accuracy. When the division ratio is 9:1, the accuracy of ours is the highest value of 99.43%, which is 0.33% higher than the GLCM + FD + Color features method [12].

Because the multi-scale triangle representation is related to K (the number of triangles), we do the following experiment (see Fig. 11). K varies from 1 to 11. When $K = 1$, the triangle representation only describes the overall layout of the leaf image, and its ability to describe details is weak. As the value of K increases, the ability of describing details increases. When $K = 7$, the accuracy and the MAP value achieve the highest. When $K > 7$ and $6 \le k \le K - 1$, $\Gamma(T^k) = \Gamma(T^6)$, the weight of the triangle representation $\Gamma(T^6)$ is getting higher and higher, the ability of Γ to describe the overall layout is relatively weak, and the accuracy is declining. When the weight of the triangle representation $\Gamma(T^k)$, $0 \le k \le 5$ is relatively lower and lower until it is negligible, the accuracy will not drop anymore. Therefore, we use $K = 7$.

TABLE 5. Accuracy of different approaches on the Flavia leaf dataset.

Descriptor	Training: Test	Accuracy (%)
Morphological Features [10]	4:1	95.42
FD+SDF+Shape Feature [2]	1:1	96.00
ACO [3]	3:1	96.25
GFPC [8]	4:1	97.00
Texture and Shape Features [9]	1:2	97.60
GLCM+FD+Color Features [12]	9:1	99.10
Our Method	1:2	98.37
	1:1	98.86
	3:1	99.21
	4:1	99.38
	9:1	99.43

3) BINARY SHAPE DATASET

In order to further evaluate the impact of KNN for optimization on the retrieval rate, we select three datasets,

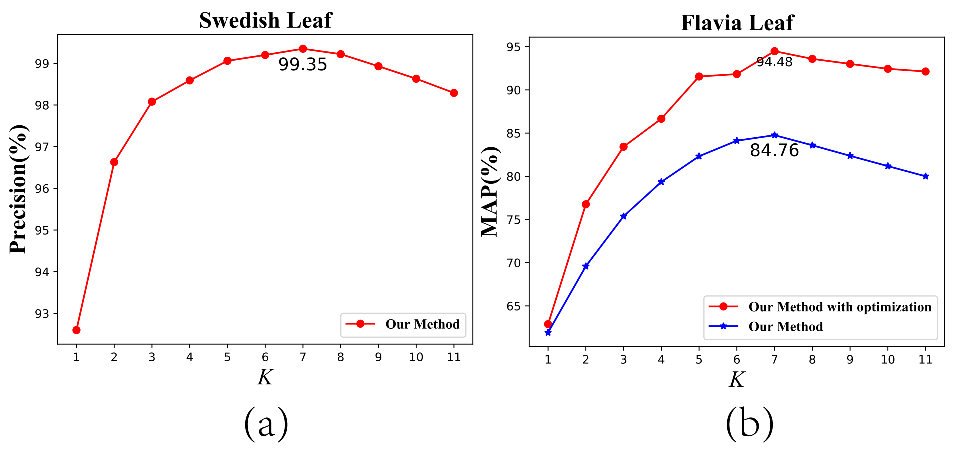

FIGURE 11. The impact of the number of different scales *K* on the proposed approach. (a) Our method tested on the Swedish Leaf Dataset. (b) Our method and Our method with optimization on the Flavia Leaf Dataset.

TABLE 6. Retrieval results (%) on Kimia99 shape dataset.

Method	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
CSR [16]	91	81	73	75	63	57	51	44	35	30
SC [17]	97	91	88	85	84	77	75	66	56	37
IDSC+DP [13]	99	99	99	98	98	97	97	98	94	79
Our Method	98	96	93	92	92	86	82	72	70	58
Our Method with Optimization	98	98	98	93	94	90	91	90	89	71

TABLE 7. Retrieval results (%) on Kimia216 shape dataset.

Method	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
CSR [16]	212	210	188	181	174	165	159	151	141	132	120
SC [17]	214	209	205	197	191	178	161	144	131	101	78
Skeleton Graph Matching [18]	216	216	215	216	213	210	210	207	205	191	177
Our Method	216	212	212	211	202	197	192	191	180	170	146
Our Method with Optimization	216	212	212	212	210	206	206	210	201	189	187

including Kimia99, Kimia216 and MPEG-7. The MPEG-7 has 70 classes, each with 20 shapes (see Fig. 8(c)). Kimia99 has 9 classes, each with 11 shapes (see Fig. 8(d)). Kimia216 has 18 classes, each with 12 shapes (see Fig. 8(e)). Same as the experimental evaluation method [13], for the Kimia99 and the Kimia216, the retrieval results come from the total number of correct matches for top 10 (in Kimia99) and top 11 (in Kimia216). Their maximum number of correct matches are 99 (in Kimia99) and 216 (in Kimia216).

The experimental evaluation of MPEG-7 is different. As same as [16], we choose "Bull's eye score", that is, each shape is queried. The percentage of correct matches in the top 40 is used as the retrieval rate for each shape and the average retrieval rate of all shapes is taken as the Bull's eye score.

Because the above three datasets are binary shape datasets and no texture information here, so we only use TUA, TVA and TSLI representations. Table 6 and Table 7 show the retrieval results of different methods on the kimia99 and kimia216. The retrieval results of our method with optimization is better than the retrieval results of our method. It shows that even if we do not use texture features, compared with the current popular methods, our method has comparable results.

We also do a comparative experiment on MPEG-7. "Bull's eye score" of our method is 78.9%, while CSR with uniform sampling [16] is 68.02%. "Bull's eye score" of our method with optimization is 86.1%, while CSR with optimal sampling is 84.17%. It demonstrates that our method has a higher accuracy than CSR. This experiment further confirms that KNN for optimization has a significant improvement in retrieval rate.

C. ROBUSTNESS

In this section, we explore robustness of our method, in which the anti-interference ability is shown in Table 8. The noise intensity S is changed from 0% to 20% (see Fig. 12). The accuracy of IDSC [13] dropped by 50.5%, and MARCH [6] dropped by 10.9 %, while our method only dropped by 5.35% in Table 7. The above experiments prove that our method has a stronger anti-interference ability.

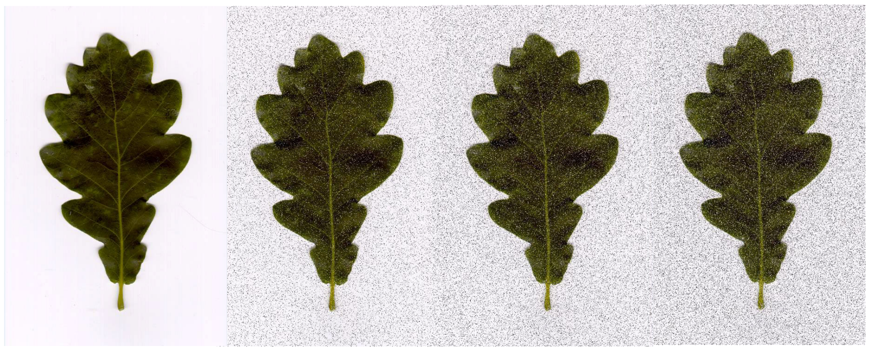

FIGURE 12. From left to right, images with 0%, 10%, 15% and 20% salt-and-pepper noise is added respectively.

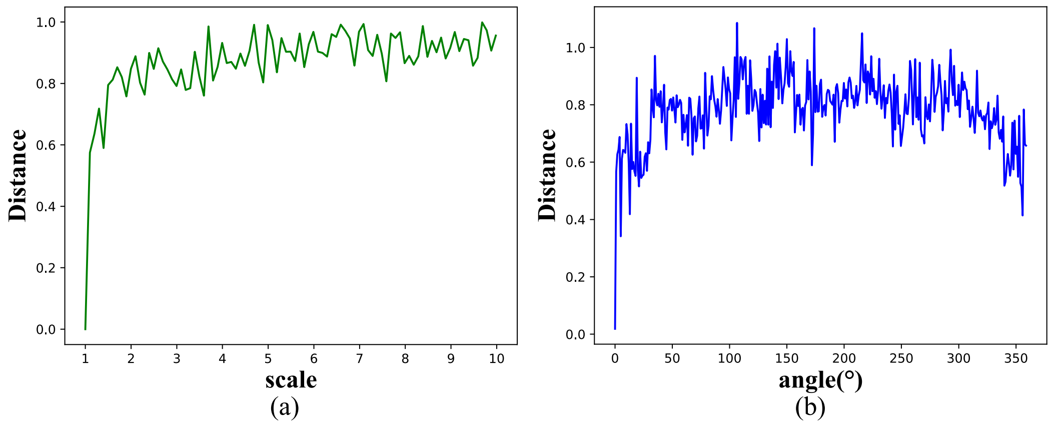

FIGURE 13. The impact of image zoom (a) and image rotation (b) on our method.

TABLE 8. Accuracy (%) on the Swedish leaf dataset with salt-and-pepper noise.

Method	S=0%	S=10%	S=15%	S=20%
IDSC [13]	94.1	73.6	52.8	43.6
MARCH [6]	96.2	93.8	90.3	85.3
Our Method	99.35	98.61	98.0	94.0

TABLE 9. Running time on different datasets.

Dataset	Feature Extraction Time (MS)	Picture recognition time (MS)	Total Time (MS)
Flavia	110.60	0.11	110.71
Flavia*	110.60	0.20	110.80
Swedish	153.90	0.07	153.97
Kimia99	23.93	0.02	23.95
Kimia99*	23.93	0.05	23.98
Kimia216	22.91	0.03	22.94
Kimia216*	22.91	0.05	22.96
MPEG-7	25.90	0.08	25.98
MPEG-7*	25.90	0.14	26.04

D. COMPUTATIONAL COST

Sections VI.*B* and VI.*C* prove the effectiveness and robustness of our method, and its efficiency will be verified in this section. The experiment calculate the feature extraction time and recognition time in five datasets. To avoid contingency, the experiment was repeated for 10 times and the average of the results was taken. It is worth noting that the leaf images on the Flavia are all 1K resolution, most of the leaf images on the Swedish Leaf Dataset are 1K resolution, a small part is 2K resolution, and the resolutions of the other three binary shape datasets are 128×128. Table 9 shows how long it takes on different datasets, where Flavia*, Kimia99*, Kimia216* and MPEG-7* is a version of Our Method with Optimization. Despite the high resolution of the leaf images, our method still cost less time to calculate. Meanwhile, although our method with optimization applies KNN for optimization, it cost little time in optimization process. Therefore, our method or our method with optimization can be applied to the real-time image recognition system.

Table 10 shows the average time consumed by different methods. Our method includes leaf contour features, texture features and shape area features. But its time is much lower than that of the method based on contour curvature feature [1], [4], [6] and the method based on mixed features [12]. Furthermore, its time consumption is 5% of the CSR and 2% of the IDSC. It is worth noting that MARCH [6] has been used

TABLE 10. Time cost of methods on the Flavia leaf dataset.

Method	Average consumption time (MS)
TSLA [1]	8.470×10^3
MDM [4]	1.780×10^2
MARCH [6]	2.130×10^2
Morphological Features [10]	1.600×10^2
GLCM+FD+Color features [12]	1.730×10^3
IDSC+DP [13]	4.920×10^3
CSR [16]	2.130×10^3
Our Method	1.107×10^2
Our Method with Optimization	1.108×10^2

in smart phones, so our method can undoubtedly be applied to the real-time leaf recognition program of smart phones.

E. THREE RST INVARIANT

In this section, our method is proved to have three RST invariant (image rotation, scale and translation are invariant). Obviously, image translation cannot change the relative position of pixels, however, image rotation and image scale cannot maintain the relative position of pixels due to the loss of numeric precision caused by the process of converting integer coordinate points. Fortunately, the influence of image rotation and image scale on our method is controllable. Fig. 13 shows how the distance between images with different scales and angles. In this experiment, the size of the image was changed from 1 to 10 times, and the angle was changed from 0° to 359°. Although the images are scaled or rotated, compared with the original image, the Euclidean distance between them is almost within 1. This error is almost caused by the loss of numeric precision. Therefore, we believe that our method is invariant to rotation, scaling and translation.

VII. CONCLUSION AND FUTURE SCOPE

In this paper, a multi-scale triangle representation is proposed, of which five representations extract contour curvature features TVA and TUA, texture features TGA and TGSD, and shape area feature TSLI from each leaf image. The multi-scale triangle representation can well describe the overall layout and local details of the leaf images, and it is also invariant to scaling, translation and rotation. Experiments show that our method and our method with optimization surpass other state-of-the-art methods in terms of recognition accuracy, retrieval rate, anti-interference ability and computational cost. We also show that KNN for optimization proposed can effectively improve the retrieval rate of the datasets. In the future, our model can be applied in mobile devices, and we are committed to researching lighter and faster leaf recognition algorithms.

REFERENCES

[1] S. Mouine, I. Yahiaoui, and A. Verroust-Blondet, "A shape-based approach for leaf classification using multiscale triangular representationm," in *Proc. 3rd ACM Int. Conf. Multimedia Retr.*, 2013, pp. 127–134.

[2] A. Aakif and M. F. Khan, "Automatic classification of plants based on their leaves," *Biosyst. Eng.*, vol. 139, pp. 66–75, Nov. 2015.

[3] M. A. Jan Ghasab, S. Khamis, F. Mohammad, and H. J. Fariman, "Feature decision-making ant colony optimization system for an automated recognition of plant species," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2361–2370, Apr. 2015.

[4] R. Hu, W. Jia, H. Ling, and D. Huang, "Multiscale distance matrix for fast plant leaf recognition," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4667–4672, Nov. 2012.

[5] X. Yu, S. Xiong, Y. Gao, Y. Zhao, and X. Yuan, "Multiscale crossing representation using combined feature of contour and venation for leaf image identification," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2016, pp. 1–6.

[6] B. Wang, D. Brown, Y. Gao, and J. L. Salle, "MARCH: Multiscale-arch-height description for mobile retrieval of leaf images," *Inf. Sci.*, vol. 302, pp. 132–148, May 2015.

[7] C. Zhao, S. S. F. Chan, W.-K. Cham, and L. M. Chu, "Plant identification using leaf shapes—A pattern counting approach," *Pattern Recognit.*, vol. 48, no. 10, pp. 3203–3215, Oct. 2015.

[8] M. Ajij, D. S. Roy, and S. Pratihar, "Plant leaf recognition using geometric features and Pearson correlations," in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Dec. 2019, pp. 1–6.

[9] J. Chaki, R. Parekh, and S. Bhattacharya, "Plant leaf recognition using texture and shape features with neural classifiers," *Pattern Recognit. Lett.*, vol. 58, pp. 61–68, Jun. 2015.

[10] M. Kumar, S. Gupta, X.-Z. Gao, and A. Singh, "Plant species recognition using morphological features and adaptive boosting methodology," *IEEE Access*, vol. 7, pp. 163912–163918, 2019.

[11] C. Yang and H. Wei, "Plant species recognition using triangle-distance representation," *IEEE Access*, vol. 7, pp. 178108–178120, 2019.

[12] M. Turkoglu and D. Hanbay, "Recognition of plant leaves: An approach with hybrid features produced by dividing leaf images into two and four parts," *Appl. Math. Comput.*, vol. 352, pp. 1–14, Jul. 2019.

[13] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

[14] M. Liu, B. C. Vemuri, S.-I. Amari, and F. Nielsen, "Shape retrieval using hierarchical total Bregman soft clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2407–2419, Dec. 2012.

[15] J. Cao, B. Wang, and D. Brown, "Similarity based leaf image retrieval using multiscale R-angle description," *Inf. Sci.*, vol. 374, pp. 51–64, Dec. 2016.

[16] G. G. Demisse, D. Aouada, and B. Ottersten, "Deformation based curved shape representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1338–1351, Jun. 2018.

[17] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[18] X. Bai and L. J. Latecki, "Path similarity skeleton graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1282–1292, Jul. 2008.

[19] K. Horaisová and J. Kukal, "Leaf classification from binary image via artificial intelligence," *Biosyst. Eng.*, vol. 142, pp. 83–100, Feb. 2016.

[20] Z. Tang, Y. Su, M. J. Er, F. Qi, L. Zhang, and J. Zhou, "A local binary pattern based texture descriptors for classification of tea leaves," *Neurocomputing*, vol. 168, pp. 1011–1023, Nov. 2015.

[21] X.-F. Wang, D.-S. Huang, J.-X. Du, H. Xu, and L. Heutte, "Classification of plant leaf images with complicated background," *Appl. Math. Comput.*, vol. 205, no. 2, pp. 916–926, Nov. 2008.

[22] O. O. Söderkvist, "Computer vision classification of leaves from Swedish trees," *Teknik Och Teknologier*, vol. 33, no. 3, pp. 202–206, 2010.

[23] 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 Int. Symp. Signal Process. Inf. Technol.*, Dec. 2007, pp. 11–16.

[24] L. J. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 424–429.

[25] T. B. Sebastian, P. N. Klein, and B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 550–571, May 2004.

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

[27] M. Shabanzade, M. Zahedi, and S. A. Aghvami, "Combination of local descriptors and global features for leaf recognition," *Signal Image Process., Int. J.*, vol. 2, no. 3, pp. 23–31, Sep. 2011.

[28] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, "Experiments of zernike moments for leaf identification," *J. Theor. Appl. Inf. Technol.*, vol. 41, no. 1, pp. 82–93, 2012.

[29] I. El Massi, Y. Es-Saady, M. El Yassa, D. Mammass, and A. Benazoun, "Automatic recognition of the damages and symptoms on plant leaves using parallel combination of two classifiers," in *Proc. 13th Int. Conf. Comput. Graph., Imag. Visualizat. (CGiV)*, Mar. 2016, pp. 131–136.

[30] P. Nijalingappa and V. J. Madhumathi, "Plant identification system using its leaf features," in *Proc. Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Oct. 2015, pp. 338–343.

[31] Y. G. Naresh and H. S. Nagendraswamy, "Classification of medicinal plants: An approach using modified LBP with symbolic representation," *Neurocomputing*, vol. 173, pp. 1789–1797, Jan. 2016.

[32] C. A. Priya, T. Balasaravanan, and A. S. Thanamani, "An efficient leaf recognition algorithm for plant classification using support vector machine," in *Proc. Int. Conf. Pattern Recognit., Informat. Med. Eng. (PRIME)*, Mar. 2012, pp. 428–432.

[33] J. C. Neto, G. E. Meyer, D. D. Jones, and A. K. Samal, "Plant species identification using elliptic Fourier leaf shape analysis," *Comput. Electron. Agricult.*, vol. 50, no. 2, pp. 121–134, Feb. 2006.

[34] E. Aptoula and B. Yanikoglu, "Morphological features for leaf based plant recognition," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1496–1499.

[35] J.-X. Du, X.-F. Wang, and G.-J. Zhang, "Leaf shape based plant species recognition," *Appl. Math. Comput.*, vol. 185, no. 2, pp. 883–893, Feb. 2007.

[36] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002, doi: 10.1109/TPAMI.2002.1017623.

[37] Y. Herdiyeni and N. K. S. Wahyuni, "Mobile application for Indonesian medicinal plants identification using fuzzy local binary pattern and fuzzy color histogram," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Depok, Indonesia, 2012, pp. 301–306.

[38] Q. K. Man, C. H. Zheng, X. F. Wang, and F. Y. Lin, "Recognition of plant leaves using support vector machine," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques* (Communications in Computer and Information Science), vol. 15, D. S. Huang, D. C. Wunsch, D. S. Levine, and K. H. Jo, Eds. Berlin, Germany: Springer, 2008, doi: 10.1007/978-3-540-85930-7_26.

[39] K. B. Lee and K. S. Hong, "An implementation of leaf recognition system using leaf vein and shape," *Int. J. Bio-Sci. Bio-Technol.*, vol. 214, no. 2, pp. 109–116, 2013.

[40] 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 Process., Int. J.*, vol. 2, no. 3, pp. 1–13, Sep. 2011.

[41] C. Sari, C. B. Akgul, and B. Sankur, "Combination of gross shape features, Fourier descriptors and multiscale distance matrix for leaf recognition," in *Proc. ELMAR*, Zadar, Croatia, 2013, pp. 23–26.

[42] J. Chaki and R. Parekh, "Plant leaf recognition using shape based features and neural network classifiers," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 10, pp. 41–47, 2011.

JIANYU SU received the bachelor's degree in computer science from South China Agricultural University. He is currently pursuing the master's degree with the Department of Computer Science, Jinan University. His research interests are machine learning, computer vision, and pattern recognition.

MEIHUA WANG received the M.S. degree from the South China University of Technology, Guangzhou, China, in 1999. She is currently an Associate Professor with South China Agricultural University, and the Director of the Data mining and Information Retrieval Laboratory. Her research interests include machine learning, computer vision, and pattern recognition.

ZHENXIN WU received the bachelor's degree in software engineering from South China Agricultural University. He is currently pursuing the master's degree with the Department of Computer Science, Jinan University. His research interests are deep learning, computer vision, and natural language processing, and he has published a paper on the image recognition.

QINGLIANG CHEN received the Ph.D. degree in computer science from Sun Yat-Sen University. He is currently a Professor with the Department of Computer Science, Jinan University. He was a Postdoctoral Fellow with Peking University, from 2010 to 2012. His research interests are machine learning and pattern recognition, and he has published more than 30 articles in peer-reviewed international journals and conferences such as AAAI, IJCAI and ECCV, and has served as the principal investigator for three research grants from National Natural Science Foundation of China.