



Contents lists available at ScienceDirect

# Journal of King Saud University – Computer and Information Sciences

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com)

## Plant leaf classification using multiple descriptors: A hierarchical approach

Jyotismita Chaki<sup>a,\*</sup>, Ranjan Parekh<sup>b</sup>, Samar Bhattacharya<sup>c</sup>

<sup>a</sup> School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Orissa, India

<sup>b</sup> School of Education Technology, Jadavpur University, Kolkata, India

<sup>c</sup> Dept. of Electrical Engineering, Jadavpur University, Kolkata, India

### ARTICLE INFO

#### Article history:

Received 6 October 2017

Revised 11 January 2018

Accepted 22 January 2018

Available online 31 January 2018

#### Keywords:

Hierarchical architecture

Feature based Shape Selection Template

Heterogeneous plant leaf

Shape

Texture

Non-green leaf

### ABSTRACT

The present work proposes another path for classification of plant species from digital leaf images. Plant leaves can have an assortment of unmistakable elements like green and non-green hue, simple and compound shape and distinctive vein designed surfaces, a solitary arrangement of elements may not be sufficiently adequate for a viable classification of heterogeneous plant sorts. A hierarchical architectural design is proposed where numerous components are joined together for a more powerful and strong classification of the visual data. The study likewise incorporates the arrangements of customization of the feature extraction modules and classifiers for best execution. The database itself is sectioned in light of conspicuous components by visual discriminators, as this enhances proficiency. As new layers can be added to the current system to take into account up to this point obscure leaves with new qualities, the design likewise provides options of adaptability. Another Feature based Shape Selection Template (FSST) is proposed for the choice of shape features for various sorts of leaves. Broad examinations are completed on two openly accessible databases including green, non-green, simple and compound leaves with variations in shape, size and designs about exhibit the advantages of the proposed strategy over best in class procedures.

© 2018 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Plants are the one of the basic component of the earth responsible for protecting the World's environment. They give sustenance, protect, medicines, fuel and keep up a sound breathable climate. Be that as it may, as of late an ever-increasing number of plants are at the very edge of termination because of ceaseless de-forestation. Thus, to monitor the plants, assembling a plant database for speedy and effective grouping and classification is an essential stride. The vast majority of these systems depend on extraction of visual components like hue, texture and shape and their portrayals as information models for correlations and classification. Albeit different parts of a plant like blossom, bud, natural

product, seed, root can be utilized for distinguishing, leaf based classification is the most widely recognized and viable approach.

A number of visual features, data modeling techniques and classifiers have been proposed for plant leaf classification. The Manifold learning based dimensionality reduction algorithm is used (Zhang et al., 2016) in plant leaf recognition as the algorithms can select a subset of effective and efficient discriminative features in the leaf images. For plant leaf recognition a dimensionality reduction method based on local discriminative tangent space alignment (LD TSA) is used where the manifold learning based dimensionality reduction algorithm is applied to reduce the size of the neighborhood matrix generated from the within class neighborhood and between class neighborhood is estimated. Deng et al. (2016) focuses on the spectral classification of weeds and crops for detecting the weeds in crop fields. The Principal Component Analysis (PCA) is used to determine the characteristic wavelengths (CW). Anjomshoae and Rahim (2016) used a template-based method for overlapping rubber tree leaf identification. Initially, the key point based feature extraction method is adopted using the Scale Invariant Feature Transform (SIFT). The steps used in the SIFT method is finding the scale space extreme by the difference of Gaussian, key point localization by principal curvature, orientation assignment

\* Corresponding author.

E-mail address: [jyotismita.chakifcs@kiit.ac.in](mailto:jyotismita.chakifcs@kiit.ac.in) (J. Chaki).

Peer review under responsibility of King Saud University.



by gradient directions and key point descriptor. An automated identification of plant species using leaf shape descriptor used by Salve et al. (2016a,b) addresses the automatic classification of plants and simplifies taxonomic classification process. In this research work, the authors use Zernike moments (ZM) and Histogram of Oriented Gradient (HOG) method as a shape descriptor. Scharr et al. (2016) compares several leaf segmentation solutions on a unique and first-of-its-kind dataset containing images from typical phenotyping experiments. Four methods are presented: three segment leaves by processing the distance transform in an unsupervised fashion and the other via optimal template selection and Chamfer matching. De Souza et al. (2016) uses the simulated annealing, differential evolution and particle swarm optimization methods, which is based on the silhouette measure, to achieve the set of optimal parameters for leaf shape characterization. The combination of texture features and shape features is used by Liu and Kan (2016) for the identification of plant leaf. Texture features are derived from local binary patterns, Gabor filters and gray level co-occurrence matrix while shape feature vector is modeled using Hu Moment invariants and Fourier descriptors. Modified Local binary patterns (MLBP) approach is used by Naresh and Nagendraswamy (2016) for classification of plant leaves based on texture features. Here mean and standard deviation of the pixels is considered instead of considering a hard threshold like normal LBP. The Angle View Projection (AVP) is used by Prasad et al. (2016) for plant leaf identification. The AVP shape profile curve (a set of four shapelets) is extracted from the leaf images. The 1-D Discrete Cosine Transform (DCT) compactness is applied over the 1-D AVP shape curve to extract the features of the leaf image. Cao et al. (2016) uses R-angle for the leaf shape characterization. R-angle describes the curvature of the contour by measuring the angle between the intersections of the shape contour with a circle of radius R centered at points sampled around the contour. Varying the parameter R of the proposed R-angle describes the notation of scale, which indicates a coarse-to-fine description of the local curvature. Visual parameters include length, width, area, perimeter is used by Sakai et al. (1996) and leaf contour shape is used by Wang et al. (2000) for the classification of plant leaves. Different data modeling techniques used include orthogonal discriminant projection (Zhang et al., 2016), the focus is on the spectral classification of weeds and crops for detecting the weeds in crop fields. The Principal Component Analysis (PCA) is used to determine the characteristic wavelengths (Deng et al., 2016), multidimensional embedding sequence similarity (Fotopoulou et al., 2013), fuzzy logic (Wang and Feng, 2002), Fourier descriptors (Yang and Wang, 2012), Recognizing leaf images based on Ring Projection Wavelet Fractal Feature is used by Wang et al. (2010), Zernike moments is used by Kadir et al. (2012) to build foliage plant identification systems. Zernike moments were combined with other features: geometric features, color moments and gray-level co-occurrence matrix (GLCM). The geometric features include aspect ratio, circularity, irregularity, solidity, convexity and two types of vein features are used. The vein features is constructed by using the morphological opening operation. The color moment features include the mean, standard deviation, skewness and kurtosis. After that the GLCM based features are extracted which include the energy, contrast, local homogeneity pair of pixels, entropy and correlation, In Salve et al. (2016a,b) an automated identification of plant species using leaf shape descriptor addresses the automatic classification of plants and simplifies taxonomic classification process. In this research uses Zernike moments (ZM) and Histogram of Oriented Gradient (HOG) method as a shape descriptor, local binary descriptors (LBD) uses (Wang et al., 2014; Le et al., 2014) used kernel descriptor (KDES) based plant leaf identification. Before the feature extraction, the leaf images are segmented using the Watershed algorithm. After that the images are converted to the grayscale

image. To extract the KDES feature, first the patch level features are extracted from the leaf images. Here three types of kernels are considered: gradient, local binary pattern and color. After extracting the patch level features, the K-means algorithm is applied to build the dictionary. After that the features are extracted from the image level using the spatial pyramid matching throughout several layers, (Markos et al., 2015) uses morphological characterization, a variety of classifiers and comparison metrics have been used viz. neural networks (Aakif and Khan, 2015; Kumar et al., 2012) describes the mobile app for identifying plant species using automatic visual recognition. First of all a binary leaf/non-leaf classifier is applied to all inputs. After that color based segmentation is applied to segment the leaf from an un-textured background. After the segmentation, the stems are removed from the binary images by the top-hat morphological operation. After the preprocessing the features are extracted by the curvature of the leaf's contour over multiple scales. The histograms of the curvature values at each scale are computed and those histograms are concatenated to form the histograms of curvature over scale feature. In Kalyoncu and Toygar (2015) the leaf image recognition is done using geometric features, multiscale distance matrix and moment invariant. Before the feature extraction segmentation of the leaf image is done by the simple adaptive threshold method over the blue channel. For the feature extraction moment invariant, convexity, perimeter ratio, multiscale distance matrix, average margin distance and the average margin peak height, peak height variance, average peak distance and peak distance variance are extracted from the contour of the leaf image. Chaki et al. (2015) uses Shape based modeling scheme based on curvelet transform and invariant moments and texture based modeling scheme based on Gabor filter and Gray Level Co-occurrence Matrix (GLCM) with neuro-fuzzy classifier.

In the vast majority of the works inspected, a solitary arrangement of elements and classifiers have been utilized to segregate between leaf classes in a given dataset of plant leaf pictures. Such an approach functions admirably when the leaf classes in the dataset are for the most part homogeneous and could be separated by a single arrangement of components. However, in Nature, plant leaves can have for all intents and purposes unbounded sorts of varieties in geometric arrangements, form shapes, hue, texture examples and additionally could be basic, compound, twisted or even fragmented (parts of the leaf missing). To take into account such heterogeneous varieties, a solitary approach is normally insufficient, rather a hierarchical architectural approach is essential where each layer utilizes a particular visual characteristic and is connected to an arrangement of custom classifiers. Results from various layers can in this manner be combined together for a more complete comprehension of the leaf features which prompts a powerful classification plot. Additionally, such designs can have arrangements for versatility, by adding new layers to take into account new leaves with various attributes.

This paper establishes the framework of such a hierarchical architecture approach for plant leaf classification and exhibits its advantages over a solitary included plan. The arrangement of the paper is as per the following area: segment 2 diagrams the proposed approach with discussions on feature calculation and order plans, segment 3 provides points of experimentation and results, segment 4 analyzes the proposed approach opposite other contemporary methodologies, while segment 5 mentions the general conclusions and degrees for future research.

## 2. Proposed approach

A block diagram depicting major functional blocks and data flow pathways of the proposed approach is shown in Fig. 1.

A query image (Q) is first subjected to a set of pre-processing operations in a pre-processing layer (P-L) before being redirected to an appropriate module in the feature extraction layer (F-L) where its visual characteristics are represented as feature vectors, which are then fed to a classifier layer (C-L) for identification and classification. The P-L consist of a pre-processing module (PP) to normalize the scaling and orientation factors of a leaf so that it can be compared with other leaves of different characteristics. The output from the PP module is a normalized query (NQ) along with a segment number (S) assigned based on the leaf aspect ratio (length by width). The normalized query is fed to a hue discriminator (C/D) which looks at its hue to identify whether the leaf is green or non-green. While most leaves are green and are difficult to identify based on their hue alone, a few leaves have striking non-green hues which can be used easily for their identification due to their dissimilarities with the multitude of green leaves. To take advantage of this fact, non-green leaves (flagged as  $CG = 0$ ) are fed to a hue module (CO) in the feature extraction layer (F-L) while green leaves ( $CG = 1$ ) are redirected along a different path for modeling their textures and shapes. The query passes along a texture discriminator (T/D) which is a user input (U I/P) or manual input, nominally inactive ( $TG = 0$ ) up to a shape discriminator (S/D) for extraction of shape features. In some cases however for leaves with prominent texture patterns and ambiguous or similar shapes, the texture discriminator can be activated ( $TG = 1$ ) to redirect the query to a texture module (TX) instead. The S/D checks whether the leaf is simple ( $SG = 0$ ) or compound ( $SG = 1$ ) and redirects it to the simple shape (SS) or compound shape (SC) modules of the feature layer. In the second step, feature vectors are generated from the color, shape and texture modules of the F-L to mathematically represent the visual features of a leaf. This work uses a 12-element hue vector  $COF(1) \dots COF(12)$ , a 20-element vector for simple shapes  $SSF(1) \dots SSF(20)$ , a 10-element compound shape vector  $SCF(1) \dots SCF(10)$  and a 144-element texture vector  $TX(1) \dots TX(144)$ . Although simple leaves have less complicated shapes than compound leaves, finer variations between them entails a larger feature vector for more reliable classification. In the third step, the feature vectors are fed to a set of custom classifiers in the classifier layer (C-L) to discriminate between various classes of leaves with different color, shape and texture properties.

The following sections provide details on each of these layers and modules. To test the effectiveness of this approach experimentations are performed on 96 classes of leaves, which includes 20 classes of non-green leaves, 42 classes of simple green leaves with prominent shapes, 24 classes of simple green leaves with prominent textures, and 10 classes of compound green leaves. Results and accuracies obtained are tabulated in the experimentations section.

#### Algorithm 1 Proposed Approach (Q)

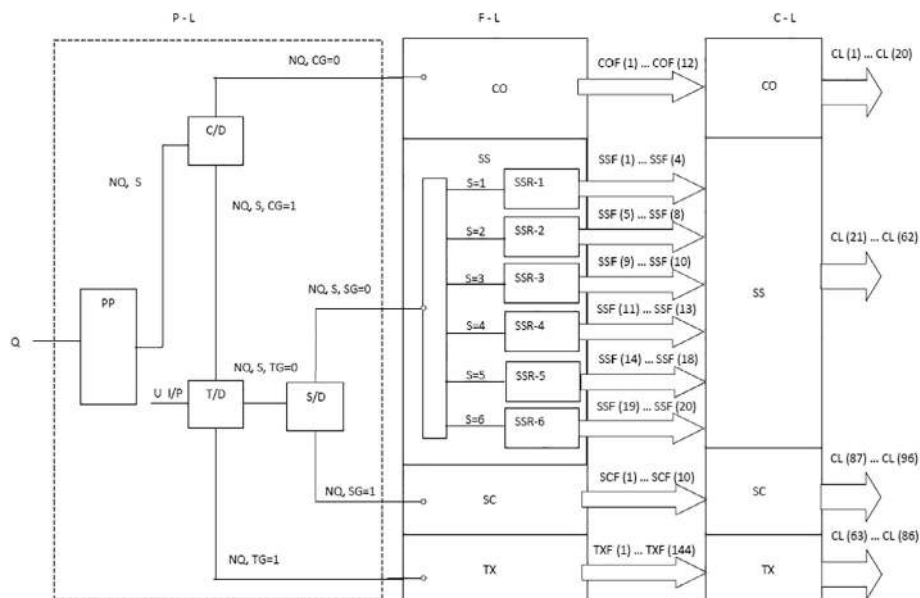
**Input Query Image:** RGB Image Q

**Output Class:**  $CL_{1 \dots 20}$  (Non-Green),  $CL_{21 \dots 62}$  (Simple Green Shape),  $CL_{63 \dots 96}$  (Compound Green Shape),  $CL_{63 \dots 86}$  (Texture)

```

1:  $NQ \leftarrow \text{pre\_process}(Q)$ 
2:  $[NOP1, NOP2] \leftarrow \text{Hue\_Discriminator}(NQ)$ 
3: if  $\text{length}(NOP2)/2 > \text{length}(NOP1)$ 
4:    $CG \leftarrow 0$ 
5: else
6:    $CG \leftarrow 1$ 
7: end if
8: if  $CG = 0$ 
9:    $COF \leftarrow F-L\_CO(NQ)$ 
10:   $CL_{1 \dots 20} \leftarrow C-L\_CO(COF)$ 
11: else
12:   $TG \leftarrow \text{Texture\_Discriminator}(NQ)$ 
13:  if  $TG = 0$ 
14:     $SG \leftarrow \text{Shape\_Discriminator}(NQ)$ 
15:    if  $SG = 0$ 
16:       $SSF \leftarrow F-L\_SS(NQ)$ 
17:       $CL_{21 \dots 62} \leftarrow C-L\_SS(SSF)$ 
18:    else
19:       $SCF \leftarrow F-L\_SC(NQ)$ 
20:       $CL_{87 \dots 96} \leftarrow C-L\_SC(SCF)$ 
21:    end if
22:  else if  $TG = 1$ 
23:     $TXF \leftarrow F-L\_TX(NQ)$ 
24:     $CL_{63 \dots 86} \leftarrow C-L\_TX(TXF)$ 
25:  end if
26: end if

```



**Fig. 1.** Hierarchical architecture of the proposed approach.

Algorithm 2 pre\_process (I)

---

**Input:** RGB image (I)  
**Output:** Normalized image (NI)  
1:  $\theta \leftarrow$  **Orientation** (I)  
2:  $I_R \leftarrow$  **Rotate** (I,  $\theta$ )  
3:  $I_{RT} \leftarrow$  **Background\_Shrunk** ( $I_R$ )  
4:  $[W, H] \leftarrow$  **size** ( $I_{RT}$ )  
5:  $AR \leftarrow W / H$   
6: **if**  $1.0 \leq AR \leq 1.9$   
7:  $NI \leftarrow I_{RT} (W/1, W)$   
8: **else if**  $2.0 \leq AR \leq 2.9$   
9:  $NI \leftarrow I_{RT} (W/1.5, W)$   
10: **else if**  $3.0 \leq AR \leq 3.9$   
11:  $NI \leftarrow I_{RT} (W/2.5, W)$   
12: **else if**  $4.0 \leq AR \leq 4.9$   
13:  $NI \leftarrow I_{RT} (W/3.5, W)$   
14: **else if**  $5.0 \leq AR \leq 5.9$   
15:  $NI \leftarrow I_{RT} (W/4.5, W)$   
16: **else if**  $6.0 \leq AR \leq 6.9$   
17:  $NI \leftarrow I_{RT} (W/5.5, W)$   
18: **end if**

---

Algorithm 3 Hue\_Discriminator (I)

---

**Input:** RGB image (I)  
**Output:** Number of Green Pixels (NOP1), Number of Non-Green Pixels (NOP2)  
1:  $I_H \leftarrow$  **rgb2hsv** (I)  
2:  $NOP1 \leftarrow 0$   
3:  $NOP2 \leftarrow 0$   
4: **for all** pixels  
5: **if**  $90 > I_H > 150$   
6:  $NOP1 \leftarrow NOP1 + 1$   
7: **else**  
8:  $NOP2 \leftarrow NOP2 + 1$   
9: **end if**  
10: **end for**

---

Algorithm 4 Background\_Shrunk (I)

---

**Input:** RGB Image (I)  
**Output:** RGB Image (B)  
1:  $[row, col] \leftarrow$  **size** (I)  
2: **for**  $p \leftarrow 1$  **to** row  
3: **if**  $I (p, :) == 0$   
4: **continue**  
5: **else**  
6:  $uppermost \leftarrow p$   
7: **break**  
8: **end if**  
9: **end for**  
10: **for**  $p \leftarrow 1$  **to** column  
11: **if**  $I (:, p) == 0$   
12: **continue**  
13: **else**  
14:  $leftmost \leftarrow p$   
15: **break**  
16: **end if**  
17: **end for**  
18: **for**  $p \leftarrow$  column **to** 1  
19: **if**  $I (:, p) == 0$

20: **continue**  
21: **else**  
22:  $rightmost \leftarrow p$   
23: **break**  
24: **end if**  
25: **end for**  
26: **for**  $p \leftarrow$  row **to** 1  
27: **if**  $I (p, :) == 0$   
28: **continue**  
29: **else**  
30:  $lowermost \leftarrow p$   
31: **break**  
32: **end if**  
33: **end for**  
34:  $B \leftarrow I (uppermost: lowermost, leftmost: rightmost)$

---

Algorithm 5 Texture\_Discriminator (I)

---

**Input:** RGB image (I)  
**Output:** Flag (TG)  
1:  $I_G \leftarrow$  **rgb2gray** (I)  
2:  $T_1 \leftarrow$  **mean** (**mean** ( $I_G$ ))  
3:  $I_{B1} \leftarrow I_G > T_1$   
4:  $I_R \leftarrow$  **Ridge\_Filter** ( $I_G$ )  
5:  $T_2 \leftarrow$  **mean** (**mean** ( $I_R$ ))  
6:  $I_{B2} \leftarrow I_R > T_2$   
7: **if** **sum** (**sum** ( $I_{B2}$ )) > **sum** (**sum** ( $I_{B1}$ )) / 4  
8:  $TG \leftarrow 1$   
9: **else**  
10:  $TG \leftarrow 0$   
11: **end if**

---

Algorithm 6 rgb2gray (I)

---

**Input:** RGB image (I)  
**Output:** Gray image ( $I_G$ )  
1: **for all** pixels  
2:  $I_G \leftarrow 0.2989 * R + 0.5870 * G + 0.1140 * B$   
3: **end for**

---

Algorithm 7 Orientation (I)

---

**Input:**  $I(x,y)$  (RGB Image of size P x Q)  
**Output:**  $\theta$  (Orientation Angle in degree)  
1:  $M_{ij} \leftarrow \sum_{x=1}^P \sum_{y=1}^Q (x)^i (y)^j f(x,y) / \sum_{x=1}^P \sum_{y=1}^Q f(x,y)$   
2:  $\mu_{ij} \leftarrow \sum_{x=1}^P \sum_{y=1}^Q (x - \mu_x)^i (y - \mu_y)^j f(x,y) / \sum_{x=1}^P \sum_{y=1}^Q f(x,y)$   
 $\mu_x$  and  $\mu_y$  is the mean of x and y component  
3:  $\theta \leftarrow \frac{1}{2} \arctan(2\mu_{11} / \mu_{20} - \mu_{02})$

---

Algorithm 8 rgb2hsv (I)

---

**Input:** I (RGB Image)  
**Output:** H (Hue Matrix of corresponding RGB)  
1:  $R \leftarrow I (:, 1)$   
2:  $G \leftarrow I (:, 2)$   
3:  $B \leftarrow I (:, 3)$   
4:  $R' \leftarrow R/255$   
5:  $G' \leftarrow G/255$

(continued on next page)

```

6:  $B' \leftarrow B/255$ 
7:  $C_{max} \leftarrow \max(R', G', B')$ 
8:  $C_{min} \leftarrow \min(R', G', B')$ 
9:  $\Delta \leftarrow C_{max} - C_{min}$ 
10: if  $\Delta = 0$ 
11:  $H \leftarrow 0^0$ 
12: end if
13: if  $C_{max} = R'$ 
14:  $H \leftarrow 60^0 \times \left( \frac{G' - B'}{\Delta} \bmod 6 \right)$ 
15: end if
16: if  $C_{max} = G'$ 
17:  $H \leftarrow 60^0 \times \left( \frac{B' - R'}{\Delta} + 2 \right)$ 
18: end if
19: if  $C_{max} = B'$ 
20:  $H \leftarrow 60^0 \times \left( \frac{R' - G'}{\Delta} + 4 \right)$ 
21: end if

```

### 2.1. Pre-processing module (PP)

A leaf picture can have varying sizes, rotational angles and translation factors because of which they initially should be standardized. The pre-processing layer (PP-L) introduces the classification framework by forcing certain rules for normalizing these elements. A leaf picture (Q) is regularly a color picture (in RGB space) arranged at an arbitrary orientation and having an irregular size, Fig. 2(a). To distinguish the rotational angle by which the leaf is slanted to the horizontal, the angle between the major axis of the leaf and the x-axis is calculated. The major axis is then turned to adjust the leaf along the level bearing, as in Fig. 2(b). To normalize the translation factors concerning the source, the foundation is contracted until the leaf just fits inside its bounding rectangle, Fig. 2(c).

A leaf picture can be of discretionary size, henceforth the system additionally requires rescaling them to pre-characterized measurements. However, since the perspective (proportion of significant hub to minor pivot) of different leaves are extraordinary, scaling them all to a solitary size can mutilate the leaf shape and influence the acknowledgment execution, particularly where shape based elements are utilized. To limit this twisting, distinctive leaf sorts are scaled to various pre-characterized sizes called “segments” contingent upon their angle proportion esteems (R). The variation of the size of leaves seen in sample leaves, six distinctive section sizes with relating names are characterized to which a leaf picture is scaled to, with next to zero mutilation. Table 1 gives points of interest of the viewpoint proportions and relating fragment numbers, names and measurements. Here  $R = 1.0$  compares to a circle while bigger esteems signify ovals with littler statures (lines) and bigger widths (sections). For computational accommodation, the width of all leaves is settled at 300 pixels. A standardized picture (NQ) from the PP module is appeared in Fig. 2(c).



Fig. 2. Pre processing steps of a leaf image (a), (b), (c).

**Table 1**  
Segments, aspect ratios and dimensions.

Segment No. (S)	Segment Name	Aspect ratio (R)	Segment Dimensions (row, col)
1	Square	$1.0 \leq R \leq 1.9$	300/1, 300
2	Very Wide	$2.0 \leq R \leq 2.9$	300/1.5, 300
3	Wide	$3.0 \leq R \leq 3.9$	300/2.5, 300
4	Medium	$4.0 \leq R \leq 4.9$	300/3.5, 300
5	Narrow	$5.0 \leq R \leq 5.9$	300/4.5, 300
6	Very Narrow	$6.0 \leq R \leq 6.9$	300/5.5, 300

### 2.2. Hue discriminator (C/D)

Hue is a noteworthy visual component utilizing which items can be recognized. The majority of the leaves are however green in hue, so separating leaves dependably in light of just hue is troublesome. However, a periodic non-green leaf may effectively be perceived by utilizing its hue data. A hue discriminator is utilized to isolate out non-green leaves from green ones. To do this the discriminator changes over the hue of the leaf from RGB to HSV hue space, as the last is outwardly more uniform i.e. measure up to increases in hue esteems prompts level with changes in visual observation. Since the essential green hue happens at 120 degrees on the hue wheel, a 30-degree precise division on either side is utilized for distinguishing green clears out. The total number of non-green ( $90^\circ \leq H \leq 150^\circ$ ) and green ( $90^\circ > H > 150^\circ$ ) pixels are counted. If number of non-green pixel/2 > number of green pixel, the leaf picture is labeled non-green (CG = 0), otherwise as green (CG = 1). Non-green leaves are sent to the hue module for recognition while green leaves are sent to the shape and texture modules for advance portrayal. The activities of the C/D can be abridged as:

$NQ(RGB) \rightarrow NQ(HSB)$

**if**  $90^\circ \leq H \leq 150^\circ$ , CG = 1

**otherwise**, CG = 0

### 2.3. Texture discriminator (T/D)

The objective of a texture discriminator is to identify leaves with prominent texture patterns or vein structure on the leaf surface, which can be used to characterize them. See Fig. 3. Simple green leaves, are fed to the T/D unit for texture discrimination.

A ridge filter is used to accentuate the ridge patterns of the leaf surface. The image is subsequently binarized and the total proportion of white pixels in the image along with an appropriate threshold is used to discriminate between textured and non-textured leaves. See Fig. 4. Leaves with high texture content are identified by setting a binary texture flag TF to 1.

The texture discriminator with a flag TG = 0 and redirects it to the shape discriminator for shape based processing. In some cases however when shape based modeling do not produce reliable results because of unclear contours or similar shapes between different leaf classes, a user input can switch the flag to TG = 1. This



Fig. 3. Non-textured vs. textured leaf.

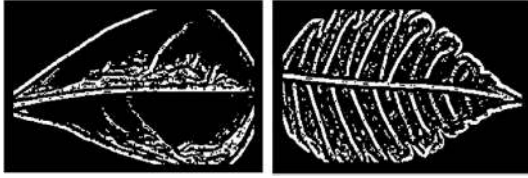


Fig. 4. Ridge patterns of non-textured and textured leaves.



Fig. 6. (a) The binary version of a compound leaf (b) After erosion.

can redirect the NQ to a texture module (TX) for texture based processing.

#### 2.4. Shape discriminator (S/D)

The first step in shape based modeling is the separation of a simple from a compound leaf. Simple leaves are characterized by a single leaf blade while compound leaves have multiple leaflets within a single unit. See Fig. 5.

A shape discriminator is utilized to isolate out the two sorts. To do this the leaf picture is first binarized by utilizing a threshold and the binary picture replaces all pixels in the picture with luminance more noteworthy than the threshold with 1 (white) and replaces every single other pixel with 0 (dark). From that point onward, a morphological erosion operation is applied to it utilizing a  $3 \times 3$  structure component made up of 1 s. Repeating this progression a fitting number of times, prompts the dynamic disintegration of the white pixels bringing about the continuous division of the leaflets from each other, as in Fig. 6. A connected component labeling utilizing 8-way connectivity is then applied to score the quantity of particular objects in the picture. If the number is 1 or less the leaf picture is labeled as simple (SG = 0), otherwise as compound (SG = 1). The quantity of times the morphological operation is repeated is streamlined in view of exploratory outcomes. Simple leaves are diverted to the SS module and compound leaves to the SC module of the component layer.

#### Algorithm 9 Shape Discriminator (I)

---

**Input:** I (RGB Image)  
**Output:** SG  
 1:  $I_G \leftarrow \text{rgb2gray}(I)$   
 2:  $I_B = I_G > \text{mean}(\text{mean}(I_G))$   
 3:  $I_E \leftarrow \text{Erode}(I_B, S)$   
 4:  $NOB \leftarrow \text{Connected\_Component}(I_E)$   
 5: **if**  $NOB \leq 1$   
 6:  $SG \leftarrow 0$   
 7: **else**  
 8:  $SG \leftarrow 1$   
 9: **end if**

---

#### Algorithm 10 Erode (I, S)

---

**Input:** I (Binary Image), S (Structuring Component)  
**Output:** E (Erosion Transformed Image)  
 1:  $E \leftarrow I \ominus S = \{c \mid c + s \in I \forall s \in S\}$

---



Fig. 5. Simple leaf vs. compound leaf.

#### Algorithm 11 Connected\_Component (I)

---

**Input:** I (Binary Image)  
**Output:** N (Number of objects)  
 1: **for all**  $I(x,y) \in I$   
 2: **if**  $I(x,y) = 0$   
 3: Proceed to the next pixel  $I(x+1, y)$   
 4: **else if**  $I(x-1, y-1)$  has a label  
 5: Assign the label to the pixel  $I(x,y)$   
 6: **else if** neither  $I(x-1, y)$  nor  $I(x, y-1)$  is not labelled  
 7: Increment label numbering and assign the latest label to  $I(x, y)$   
 8: **else if**  $I(x-1, y) \oplus I(x, y-1)$  is labelled  
 9: Assign the label to  $I(x,y)$   
 10: **else if** both  $I(x-1, y)$  and  $I(x, y-1)$  are labelled  
 11: Assign the label of  $I(x-1, y)$  to  $I(x, y)$   
 12: Record the equivalence if labels of  $I(x-1, y)$  and  $I(x, y-1)$  are not identical  
 13: **end if**  
 14: **end for**  
 15:  $N \leftarrow \text{max}(\text{max}(\text{label of } I))$

---

#### 2.5. Hue module (CO)

The target of the hue module is to extricate hue data from a leaf picture which can be utilized for its order. The hue of a non-green leaf is demonstrated by disintegrating into the H, S, V channels. To limit hue varieties over the leaf surface, the surface is partitioned into four quadrants and standard deviations of the segment channels over every quadrant is linked to have the 12-component hue highlight vector (COF) as appeared in Eq. (1). The subscripts 1 to 4 assign the quadrant number. Fig. 7 delineates the procedure.

$$\text{COF} = \{a, b, c\} \quad (1)$$

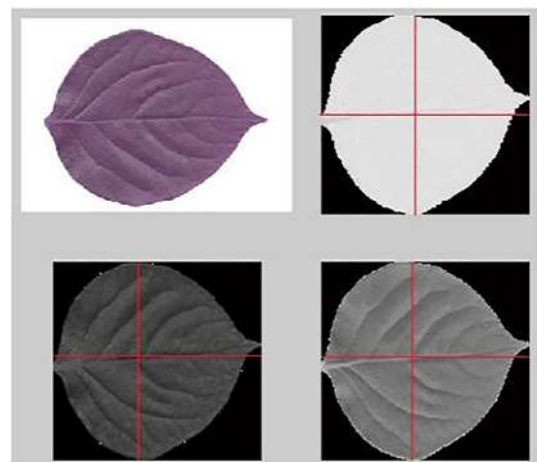


Fig. 7. A non-green leaf and its decomposition into H (top-right), S (bottom-left) and V (bottom right) channels.

where

$$a = \{\sigma(H1), \sigma(H2), \sigma(H3), \sigma(H4)\}$$

$$b = \{\sigma(S1), \sigma(S2), \sigma(S3), \sigma(S4)\}$$

$$c = \{\sigma(V1), \sigma(V2), \sigma(V3), \sigma(V4)\}$$

## 2.6. Simple shape module (SS)

The goal of the SS module is to extract shape data from a simple leaf picture which can be utilized for its characterization. Shape based demonstrating of a simple leaf is instated by breaking down how shape highlights differ with shapes i.e. which features are upgraded for which shapes. As a major aspect of the pre-processing operation, leaves have been isolated into 6 sections in light of their aspect ratio (R). Some essential shape extraction techniques are chosen and another Feature based Shape Selection Template (FSST) rule is developed for each segment as various sorts of leaf shapes may require distinctive arrangement of shape for characterization. An arrangement of four fundamental shape parameters in particular significant pivot length (M), minor hub length (N), leaf region (A), leaf form edge (P) alongside six inferred parameters, E (randomness), F (frame factor), G (proportion of edge to real hub), R (angle proportion), S (proportion of border to the sum of principal axes), T (rectangularity), are utilized for shape representation. These are characterized below.

$$E = \sqrt{1 - (N/M)^2} \quad (2)$$

$$F = 4\pi A/P^2 \quad (3)$$

$$G = P/M \quad (4)$$

$$R = M/N \quad (5)$$

$$S = P/(M + N) \quad (6)$$

$$T = M * N/A \quad (7)$$

To lead the enhancement investigation, a FSST is produced where leaves are represented by their nearest geometrical partner i.e. ovals. The aspect ratio R of the ovals is fluctuated from 1.0 to 6.9 in increments of 0.1 to create 60 tests clubbed into 6 sections according to Table 1. See Fig. 8. Each line in the figure shows a section and every segment portrays varieties inside a segment. As the aspect ratio expands, the presence of the ovals inside each section look very comparable as is apparent from the last 3 rows in the figure. This legitimizes the clubbing of higher aspect ratio over 7 into a similar segment 6.

For every oval, the accompanying elements are figured: A, E, F, G, P, S, T. To characterize the ellipses, each section is dealt with as a class and the ovals inside a fragment as the examples inside each class. It is clear that last couple of ovals of each fragment are inalienably like the initial couple of ovals of the following portion. To consider the partial overlap between the classes, a fuzzy classifier is utilized which produces weight factors for each specimen demonstrative of its participation over all classes. Experimentation demonstrate best grouping outcomes are created if the first and last specimens are utilized for preparing and the rest of the 8 tests for testing. The consequences of grouping are utilized to produce the accompanying upgraded feature set for each class which is from this time onward alluded to as rules:

The SS module utilizes the fragment number S, obtained from the PP module, to redirect approaching leaf pictures into 6 sub-modules SSR-1 to SSR-6, according to Table 1. These sub-modules figure elements to create a 20-component highlight vector SSF (1) to SSF (20) according to Table 2. These standards are from this time onward alluded as FSST rules.

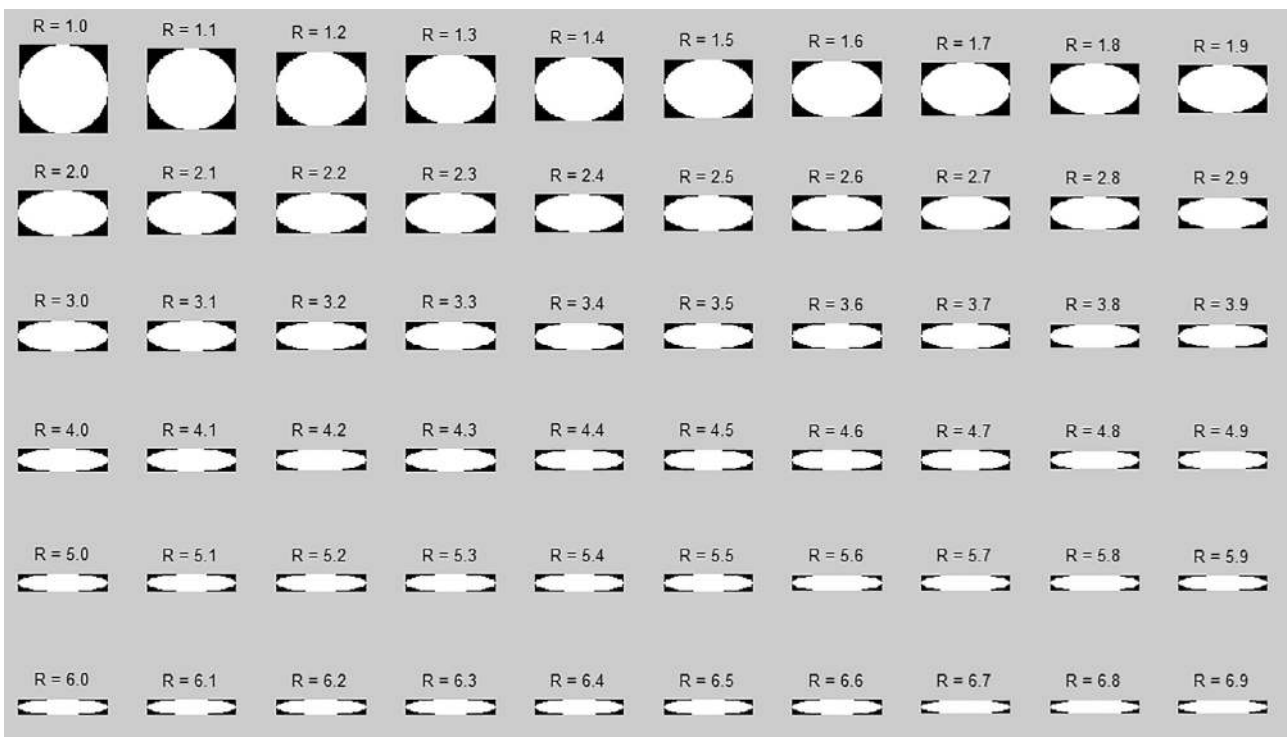


Fig. 8. Ellipses of varying aspect ratios used for generating the Feature based Shape Selection Template (FSST) and conducting analysis tests.

**Table 2**  
Rules & segments.

Rule	Feature
Rule 1 for Square Segment	A, E, F, G
Rule 2 for Very Wide Segment	A, E, P, T
Rule 3 for Wide Segment	P, T
Rule 4 for Medium Segment	A, E, S
Rule 5 for Narrow Segment	A, P, F, G, T
Rule 6 for Very Narrow Segment	A, E

2.7. Compound shape module (SC)

A compound leaf has attributes which tend to change along its shape. To characterize the variety, a compound leaf is separated into five segments along the significant pivot, named P1 to P5. See Fig. 9

For each parcel four elements are figured viz. real pivot length (M), minor-hub length (N), leaf zone (A), leaf border (P) and other inferred highlights as follows: R, T, S, F, G. The tenth element named progressive centroid (H) is ascertained by utilizing a recursive system to separate each partition and processing the outright total of the x-coordinate of the centroids. The 10 features constitute the partition shape vector (U<sub>i</sub>) for the i-th segment.

$$U_i = M, N, A, P, R, T, S, F, G, H$$

The partition vector for each partition is added up to form the 10-element feature vector for the entire leaf (SCF).

$$SCF = \{U_{P1} + U_{P2} + U_{P3} + U_{P4} + U_{P5}\} \tag{8}$$

2.8. Texture module (TX)

The objective of the texture module is to extract texture information from a leaf surface which can be used for its classification. A fuzzy hue and texture descriptor is used where hue information is combined with texture to improve recognition accuracies. The procedure is mentioned in brief in the following sections for the convenience of the reader.

2.8.1. Fuzzy hue descriptor

The hue descriptor is generated using a 3-step process:

Step 1: Generation of 8 hue areas

Hue of a picture is communicated in the HSV (Hue-Saturation-Value) space. Channel H is isolated into 8 fuzzy zones in view of the reaction of an arrangement of Coordinate Logic Filter (CLF) on extraordinarily built counterfeit pictures. These zones are: (0) Red to Orange, (1) Orange, (2) Yellow, (3) Green, (4) Cyan, (5) Blue, (6) Magenta and (7) Blue to red.

Step 2: Generation of 10-bin hue histogram

Channel S is isolated into two fuzzy zones named as 0 and 1, while channel V is partitioned into three fuzzy regions named as 0, 1 and 2. In light of an arrangement of fuzzy derivation administrators, the above hues are consolidated to produce a 10-container histogram, as determined underneath. Table 3 shows the Bin information

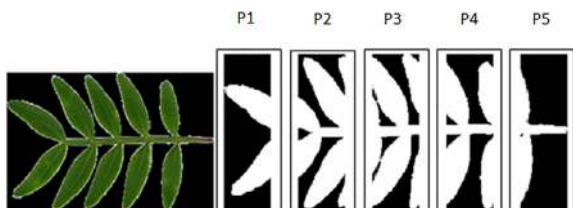


Fig. 9. Processing a compound green leaf.

**Table 3**  
Hue, S, V bin information.

Hue	Channel S	Channel V	Hue_Bin
~	~	0	Black
~	0	1	Gray
~	0	2	White
Red to Orange	1	1 or 2	Red
Orange	1	1 or 2	Orange
Yellow	1	1 or 2	Yellow
Green	1	1 or 2	Green
Cyan	1	1 or 2	Cyan
Blue	1	1 or 2	Blue
Magenta	1	1 or 2	Magenta
Magenta to red	1	1 or 2	Red

information obtained from different values of Hue, S and V. ~ symbol represents any value.

The output of this stage is a 10-bin histogram, where each bin represents a preset hue viz. (0) Black, (1) Gray, (2) White, (3) Red, (4) Orange, (5) Yellow, (6) Green, (7) Cyan, (8) Blue and (9) Magenta.

Step 3: Generation of 24-bin hue histogram

In the third stage, two separate fuzzy membership function partitions the S and V channels into 2 fuzzy areas each assigned as 0 and 1. Three fuzzy induction rules assign a hue as “unadulterated” if S = 1 and V = 1, a hue as “light” if S = 0 and V = 1, and a hue as “dull” if s = 0 or 1 and V = 0. Utilizing these derivations governs the 10-bin histogram is extended to a 24-bin histogram as takes after:

(a) If the input hue corresponds to bins 0, 1 or 2 (i.e. black, gray, white),

then it is represented as it is in the output bins

(b) If the input hue corresponds to bins 3 to 9, then inference rules separates out each hue into three variants: pure, light, dark

The yield of the third stage is the era of a 24-bin hue histogram as beneath: (0) dark, (1) Gray, (2) White, (3) Dark red, (4) Red, (5) Light red, (6) Dark orange, (7) Orange, (8) Light orange, (9) Dark yellow, (10) Yellow, (11) Light yellow, (12) Dark green, (13) Green, (14) Light green, (15) Dark cyan, (16) Cyan, (17) Light cyan, (18) Dark blue, (19) Blue, (20) Light blue, (21) Dark fuchsia, (22) Magenta, (23) Light red.

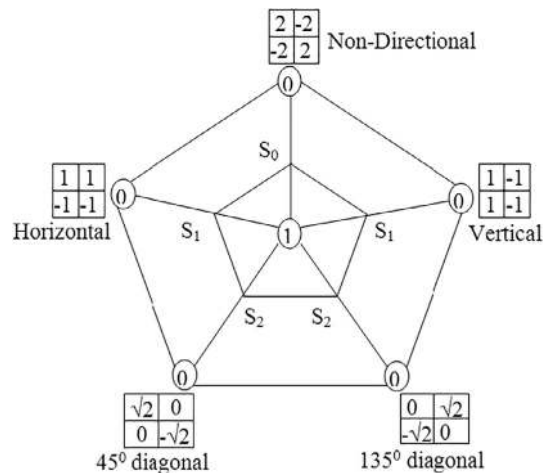


Fig. 10. Edge type diagram.



### 2.8.2. Texture descriptor

The texture descriptor is generated using the five filters of MPEG-7 Edge Histogram Descriptor (EHD). These are used to define the edges as vertical, flat, 45° corner to corner, 135° inclining and non-directional. The edge type diagram is shown in Fig. 10.

The whole picture is separated into various picture squares. Each picture piece is again sub-isolated into 4 sub squares. Let  $a_0, a_1, a_2, a_3$  be the normal dim levels of the 4 sub-squares of a particular picture piece. The channel coefficients for vertical, flat, 45° corner to corner, 135° inclining and non-directional edges are assigned as:  $F_v, F_h, F_{45d}, F_{135d}$  and  $F_{nd}$ . For the  $(i,j)$ -th picture obstruct the edge extents figured from its constituent sub-pieces are characterized as takes after:

$$E_v(i,j) = |a_0(i,j) * F_v(0) + a_1(i,j) * F_v(1) + a_2(i,j) * F_v(2) + a_3(i,j) * F_v(3)| \quad (9)$$

$$E_h(i,j) = |a_0(i,j) * F_h(0) + a_1(i,j) * F_h(1) + a_2(i,j) * F_h(2) + a_3(i,j) * F_h(3)| \quad (10)$$

$$E_{45d}(i,j) = |a_0(i,j) * F_{45d}(0) + a_1(i,j) * F_{45d}(1) + a_2(i,j) * F_{45d}(2) + a_3(i,j) * F_{45d}(3)| \quad (11)$$

$$E_{135d}(i,j) = |a_0(i,j) * F_{135d}(0) + a_1(i,j) * F_{135d}(1) + a_2(i,j) * F_{135d}(2) + a_3(i,j) * F_{135d}(3)| \quad (12)$$

$$E_{nd}(i,j) = |a_0(i,j) * F_{nd}(0) + a_1(i,j) * F_{nd}(1) + a_2(i,j) * F_{nd}(2) + a_3(i,j) * F_{nd}(3)| \quad (13)$$

The maximum of the magnitudes is calculated from the above edge magnitudes

$$E_{max} = \max (E_v, E_h, E_{45d}, E_{135d}, E_{nd}) \quad (14)$$

This is subsequently used to normalize the magnitudes

$$E'_v = E_v/E_{max},$$

$$E'_h = E_h/E_{max},$$

$$E'_{45d} = E_{45d}/E_{max},$$

$$E'_{135d} = E_{135d}/E_{max},$$

$$E'_{nd} = E_{nd}/E_{max}$$

The framework characterizes each Image Block in a two-stage process: to begin with, the framework figures the maximum esteem. The maximum esteem must be more noteworthy than the characterized limit for the Image Block to be delegated a Texture Block; else it is named a Non-Texture Block (Linear). In the event that the Image Block is named a Texture Block, every  $E'$  esteem is put on the pentagonal outline of Fig. 8 along the line relating to advanced channel from which it was figured. The graph's middle compares to esteem 1 and the external edge relates to esteem 0. If any  $m$  esteem is more noteworthy than the edge on hold where it takes an interest, the Image Block is characterized into the specific sort of edge. In this way, an Image Block can take an interest in more than one edge sort.

The yield of this stage is portrayal of a picture square utilizing a 6-canister surface histogram: non-edge, non-directional edge, flat edge, vertical edge, 45° corner to corner, 135° inclining.

### 2.8.3. Combined descriptor

For an info picture  $NQ$ , the fuzzy hue descriptor initially produces a 24-bin hue descriptor in light of the hue hue contained in the picture. For each receptacle, the surface descriptor produces a 6-bin texture descriptor to evaluate the texture introductions.

The consolidated descriptor in this manner creates a  $24 \times 6 = 144$  component feature vector  $TX(1)$  to  $TX(144)$ .

### 2.9. Classification layer

The  $i$ -th leaf class is portrayed by a gathering of  $n$  part pictures, isolated into preparing and testing tests. Amid the training stage, the 12-component hue, the 20-component simple shape, the 10-component compound shape and 144-component texture vectors

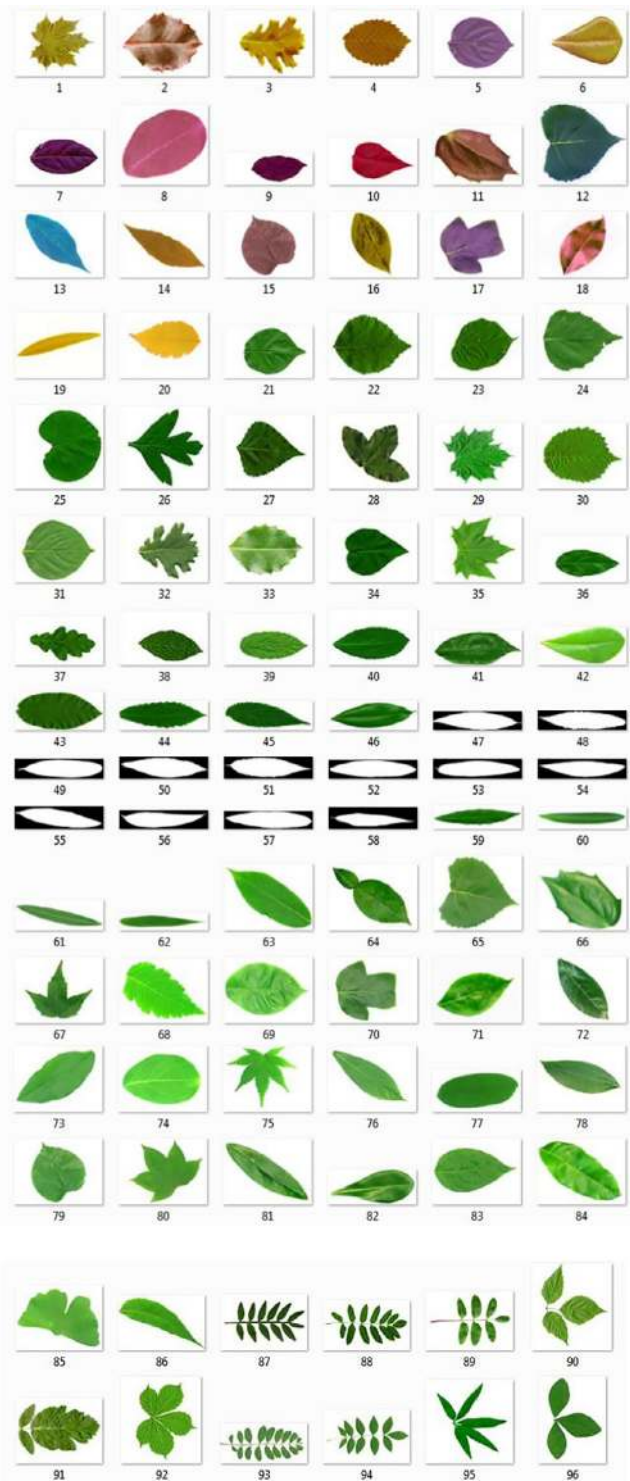


Fig. 11. Samples of 96 classes of the dataset.

are registered from the training samples. The components are put away in a plant database (PDB) fragmented into the comparing parts: hue, shape (simple), shape (compound) and surface. This guarantees correlations amid the testing stage will just happen inside one of these subsets as dictated by a proper discriminator. The flowline portrayed in Fig. 1 is utilized for characterization of a test. Characterization for both the hue and shape modules is finished utilizing a neuro-fuzzy classifier (NFC) in view of a scaled conjugate inclination calculation. The inspiration of utilizing a neural classifier emerges from the way that like most example acknowledgment issues, there is no settled numerical model in view of which information tests could be characterized, rather it should be done exclusively on the premise of a non-straight mapping between an arrangement of info and yield perceptions. Since a specific leaf test can have similitude with numerous classes, a fluffy classifier is utilized as it represents probabilities

of a specimen for having a place with a few classes not at all like a neural system which forces elite characterization. In view of the test leaf sort the seeking procedure dependably happens inside a fitting subset of the aggregate database, which improves the computational load. For the arrangement of the surface elements, the Euclidean separation is utilized, since a 144-component vector on a neural classifier can conceivably expand union time and computational load.

### 3. Experimentations and results

#### 3.1. Dataset

Experimentation are finished utilizing 1920 leaf pictures isolated into 96 classes, gathered from [Plantscan dataset](#) and [Flavia dataset](#). Out of 20 pictures for every class, 10 are utilized for preparing and 10 for testing. Fig. 11 demonstrates tests of each class in the data set. Classes 1 to 20 contain non-green leaves, classes 21 to 62 contain basic leaves which are recognized by their shapes, classes 63 to 86 contain basic leaves distinguished by their texture, and classes 87 to 96 contain compound leaves.

#### 3.2. Preprocessing

The pre-processing step involves the sample images rescaled to standard segment sizes. Table 4 depicts the segment numbers of simple leaves belonging to classes 21 to 62.

#### 3.3. Recognition of non-green leaves

The variation of color features defined in Eq. (1) over the 20 classes (class 1 to 20) is shown in Fig. 12. The H, S, V components of the feature, averaged of all training samples for each class, are depicted.

Table 5 shows recognition results for non-green leaves which includes class 1 to class 20 i.e. 20 classes with 10 test samples per class (200 samples). Overall accuracy is 100%. Fig. 13 shows the NFC class label outputs for 200 test samples.

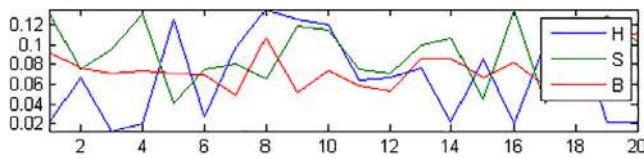
#### 3.4. Recognition of simple leaves using shape features

Prior to the acknowledgment of simple green leaves the pictures are diverted to one of the six sub-modules SSR-1 to SSR-6 in view of their section number S. The current dataset contains 15 classes for Square section (21 to 35), 8 classes for Very Wide (36 to 43), 3 classes for Wide (44 to 46), 5 classes for Medium (47 to 51), 8 classes for Narrow (52 to 59) and 3 classes for Very Narrow (60 to 62). See Table 3.

Square Segment is represented by FSST Rule-1 which incorporates four features: A, E, F, G, effective over 15 classes (21 to 35). Fig. 14 demonstrates the variety of these features arrived at the average for each of the classes.

**Table 4**  
Segments & classes.

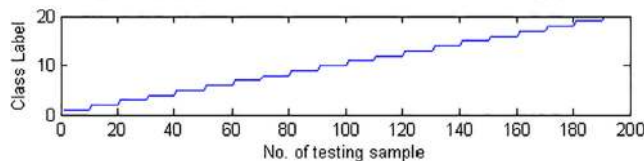
Class	Segment	Class	Segment	Class	Segment
21–35	1	36–43	2	44–46	3
47–51	4	52–59	5	60–62	6



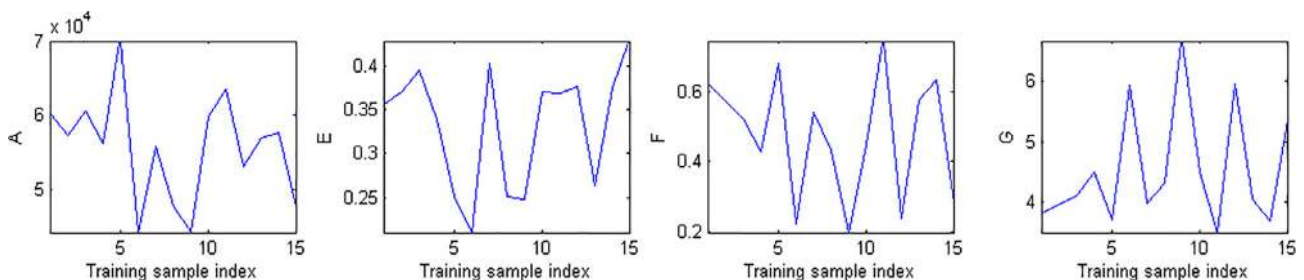
**Fig. 12.** Variation of color features over 10 classes of non-green leaves.

**Table 5**  
Percent recognition rates for non-green leaves.

Class	Acc	Class	Acc	Class	Acc	Class	Acc	Class	Acc
1	100	2	100	3	100	4	100	5	100
6	100	7	100	8	100	9	100	10	100
11	100	12	100	13	100	14	100	15	100
16	100	17	100	18	100	19	100	20	100



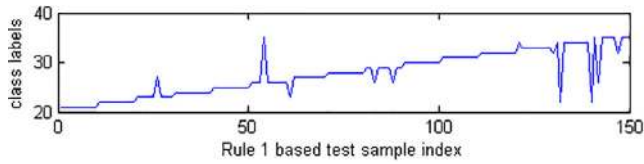
**Fig. 13.** NFC based class output labels for 200 non-green test samples.



**Fig. 14.** Variation of features over classes in square segment.

**Table 6**  
Percent recognition rates for square segment.

Class	Acc	Class	Acc	Class	Acc	Class	Acc	Class	Acc
21	100	22	100	23	90	24	100	25	100
26	90	27	90	28	100	29	80	30	100
31	100	32	100	33	80	34	80	35	80



**Fig. 15.** NFC based class output labels for 150 simple green test samples belonging to Square segment.

The performance results using NFC are in Table 6. Overall accuracy is 92.7%. Fig. 15 shows the NFC class label outputs for 150 test samples.

Very Wide Segment is represented by FSST Rule-2 which incorporates four features: A, E, F, T, effective over 8 classes (36 to 43). Fig. 16 demonstrates the variety of these features averaged for each of the classes.

The performance results using NFC are in Table 7. Overall accuracy is 88.8%. Fig. 17 shows the NFC class label outputs for 80 test samples.

Wide Segment is represented by FSST Rule-3 which incorporates two features: P, T, effective over 3 classes (44 to 46). Fig. 18 demonstrates the variety of these features averaged for each of the classes.

The performance results using NFC are in Table 8. Overall accuracy is 100%. Fig. 19 shows the NFC class label outputs for 30 test samples.

Medium Segment is represented by FSST Rule-4 which incorporates three features: A, E, S effective over 5 classes (47 to 51). Fig. 20 demonstrates the variety of these features averaged for each of the classes.

The performance results using NFC are in Table 9. Overall accuracy is 94%. Fig. 21 shows the NFC class label outputs for 50 test samples.

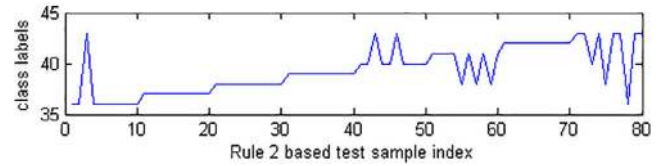
Narrow Segment is represented by FSST Rule-5 which incorporates five features: A, P, F, G, T effective over 8 classes (52 to 59). Fig. 22 demonstrates the variety of these features averaged for each of the classes.

The performance results using NFC are in Table 10. Overall accuracy is 83.8%. Fig. 23 shows the NFC class label outputs for 80 test samples.

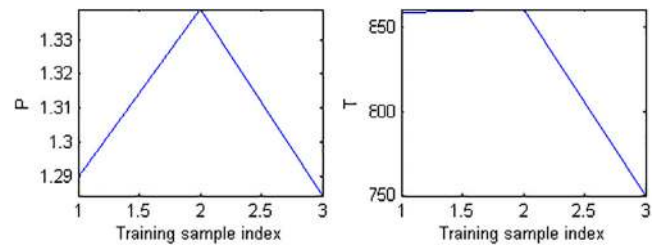
Very narrow Segment is represented by FSST Rule-6 which incorporates two features: A, E effective over 3 classes (60 to 62).

**Table 7**  
Percent recognition rates for very wide segment.

Class	Acc	Class	Acc	Class	Acc	Class	Acc	Class	Acc
36	90	37	100	38	100	39	100	40	80
41	70	42	100	43	70				



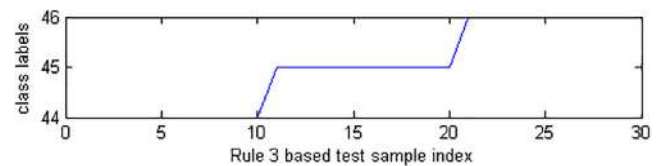
**Fig. 17.** NFC based class output labels for 80 simple green test samples belonging to Very Wide segment.



**Fig. 18.** Variation of features over classes in Wide segment.

**Table 8**  
Percent recognition rates for wide segment.

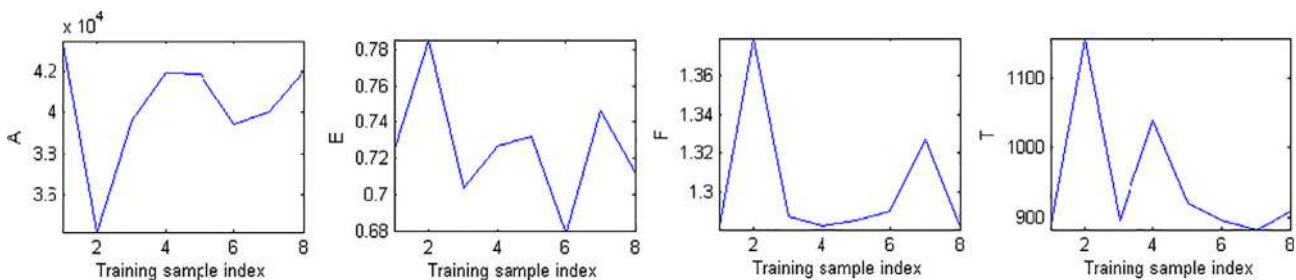
Class	Acc	Class	Acc	Class	Acc
44	100	45	100	46	100



**Fig. 19.** NFC based class output labels for 30 simple green test samples belonging to Wide segment.

Fig. 24 demonstrates the variety of these features averaged for each of the classes.

The performance results using NFC are in Table 11. Overall accuracy is 100%. Fig. 25 shows the NFC class label outputs for 30 test samples.



**Fig. 16.** Variation of features over classes in Very Wide segment.

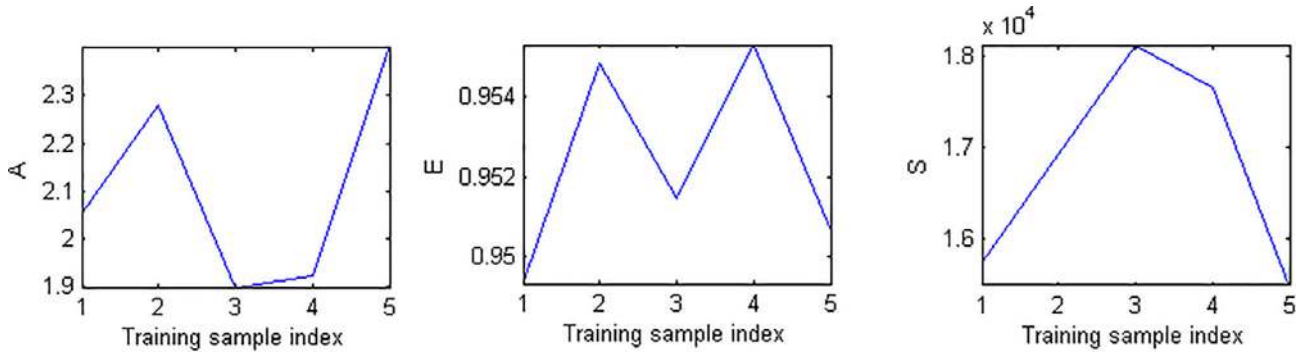


Fig. 20. Variation of features over classes in Medium segment.

Table 9

Percent recognition rates for medium segment.

Class	Acc	Class	Acc	Class	Acc	Class	Acc	Class	Acc
47	100	48	100	49	90	50	90	51	90

Table 10

Percent recognition rates for narrow segment.

Class	ACC	Class	Acc	Class	Acc	Class	Acc	Class	Acc
52	60	53	80	54	80	55	100	56	90
57	80	58	100	59	80				

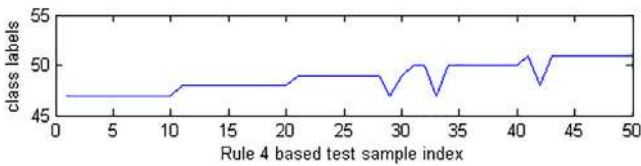


Fig. 21. NFC based class output labels for 50 simple green test samples belonging to Medium segment.

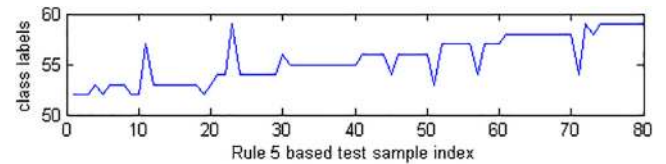


Fig. 23. NFC based class output labels for 80 simple green test samples belonging to Narrow segment.

The overall accuracy for 42 classes of simple green leaves is  $(92.7 + 88.8 + 100 + 94 + 83.8 + 100)/6 = 93.2\%$ .

To cross-check whether the picked features for each section gives ideal execution, each list of capabilities is connected over all portions and exactness comes about are looked at. Table 12 demonstrates the similar precision. It demonstrates that for Square Segment, the best outcomes are created by Rule-1 (92.7%), for Very Wide section, the best outcomes are delivered by Rule-2 (88.8%), for Wide portion, the best outcomes are created by Rule-3 (100%), for Medium segment, the best outcomes are delivered by Rule-4 (94%), for Narrow section, the best outcomes are delivered by Rule-5 (83.8%), for Very Narrow segment, the best outcomes are created by Rule-6 (100%). Fig. 26 demonstrates a plot of how different rules (along the level pivot) deliver percent exactness for each section.

This gives test affirmation of the way that rules and capabilities when tweaked in view of leaf sort have a tendency to give

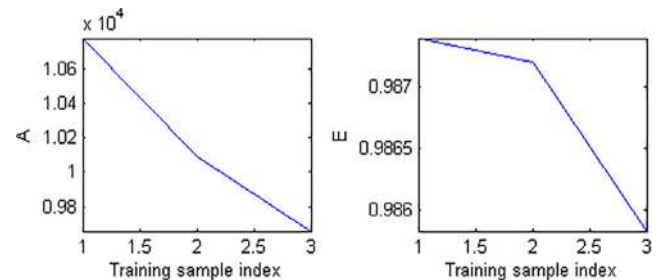


Fig. 24. Variation of features over classes in Very Narrow segment.

Table 11

Percent recognition rates for very narrow segment.

Class	ACC	Class	Acc	Class	Acc
60	100	61	100	62	100

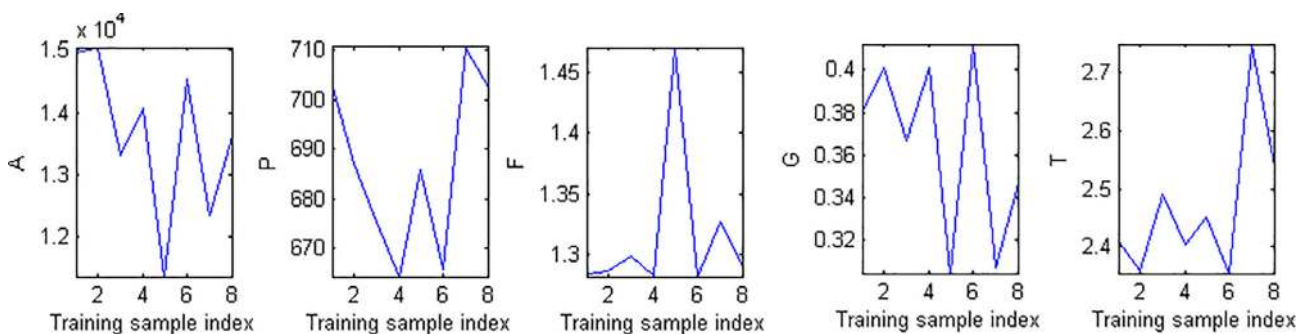


Fig. 22. Variation of features over classes in Narrow segment.

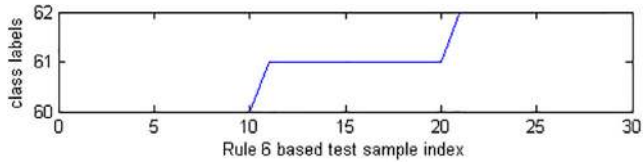


Fig. 25. NFC based class output labels for 30 simple green test samples belonging to Very Narrow segment.

Table 12

Cross validation of recognition rates.

Feature	Square	Very Wide	Wide	Medium	Narrow	Very Narrow
SSR1	92.7	55	93.3	92	62.5	96.7
SSR2	91.3	88.8	96.7	82	76.3	96.7
SSR3	85.3	62.5	100	70	61.3	96.7
SSR4	83.3	76.3	93.3	94	62.5	100
SSR5	91.3	70	96.7	86	83.8	90
SSR6	41.3	71.3	90	76	56.3	100

preferred outcomes over when a solitary arrangement of components is connected for each leaf sort. This is the reason for the enhanced execution of the various leveled architecture.

3.5. Recognition of compound leaves

Green compound leaves are characterized using a 10-element vector applied over the 10 classes (87 to 96). Fig. 27 shows the variation of these features averaged for each of the classes.

The performance results using NFC are in Table 13. Overall accuracy is 94%. Fig. 28 shows the NFC class label outputs for 100 test samples.

Table 13

Percent recognition rates for compound leaves.

Class	Acc	Class	Acc
87	100	92	60
88	100	93	100
89	100	94	100
90	100	95	70
91	100	96	100

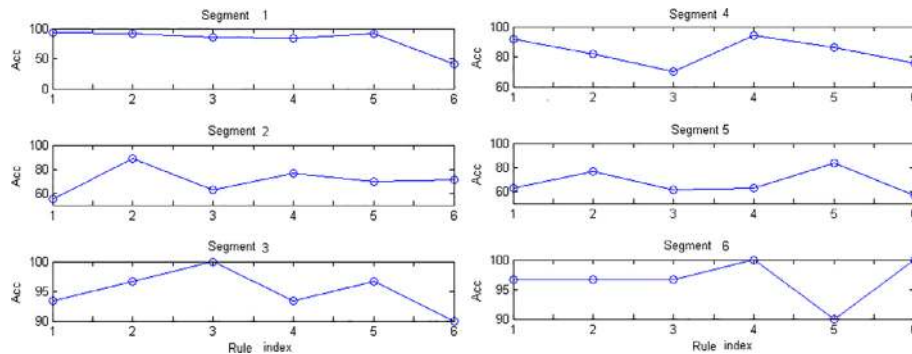


Fig. 26. Variation of accuracy for each rule applied to each segment.

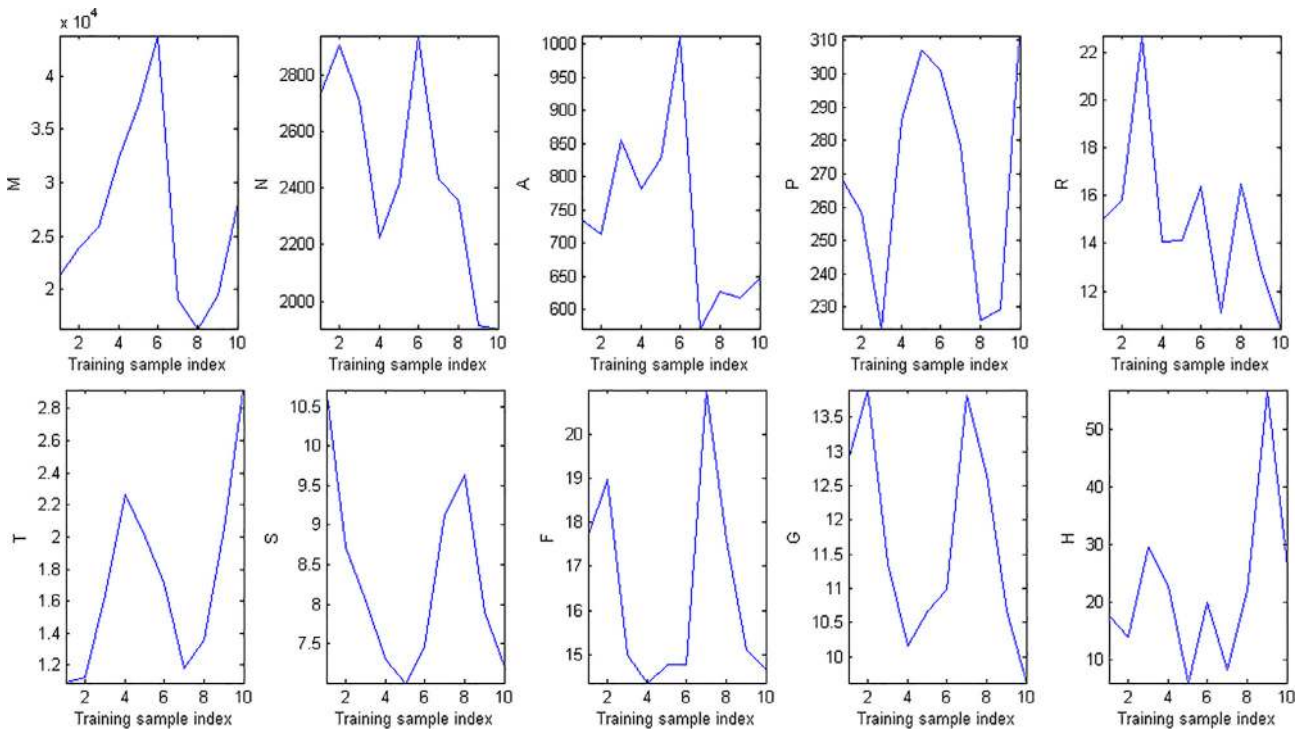


Fig. 27. Variation of features over 10 classes.

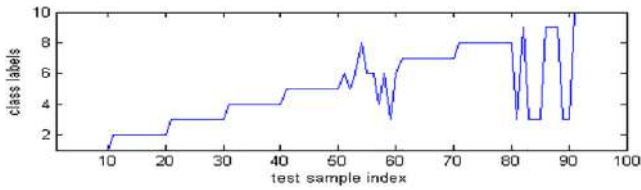


Fig. 28. NFC class output labels for 100 compound green test samples.

### 3.6. Recognition of simple leaves using texture features

Leaves whose diagram shapes are not noticeable i.e. can't be demonstrated with adequate precision, or not one of a kind i.e. bearing close likenesses with different classes, can't be dependably separated utilizing shape feature. For such leaves texture based components utilizing combined descriptors are utilized for grouping, as clarified previously. Fig. 29 demonstrates the varieties of the 144-component highlight vectors found the middle value of for 10 classes.

The precision got by applying these features over 24 classes (63 to 86) are organized in Table 14. The general exactness for 24 classes is 95.4% acquired by utilizing Euclidean separations as order metric.

## 4. Analysis

The present work is contrasted with other contemporary works, by applying a portion of the methodologies found in surviving writing to the current dataset to watch their exhibitions. It should be specified here that a large portion of these current methodologies have been intended for basic green leaves which would represent their low classification rates when connected to non-green takes off. As the current methodologies are for the classification of the simple leaves, the correlation is done just with the basic leaf pictures of aggregate 86 classes and the last 10 classes are excluded in the examination.

In Aakif and Khan (2015) leaf shapes are displayed utilizing Fourier descriptors, morphological components and shape characterizing features. For the current dataset, it was discovered that the smooth sinusoidal premise capacities were unacceptable for displaying the transient signs happening along sharp leaf forms. Likewise, as the blend of above said three sorts of components (add up to six elements) are utilized for arrangement, there can be an

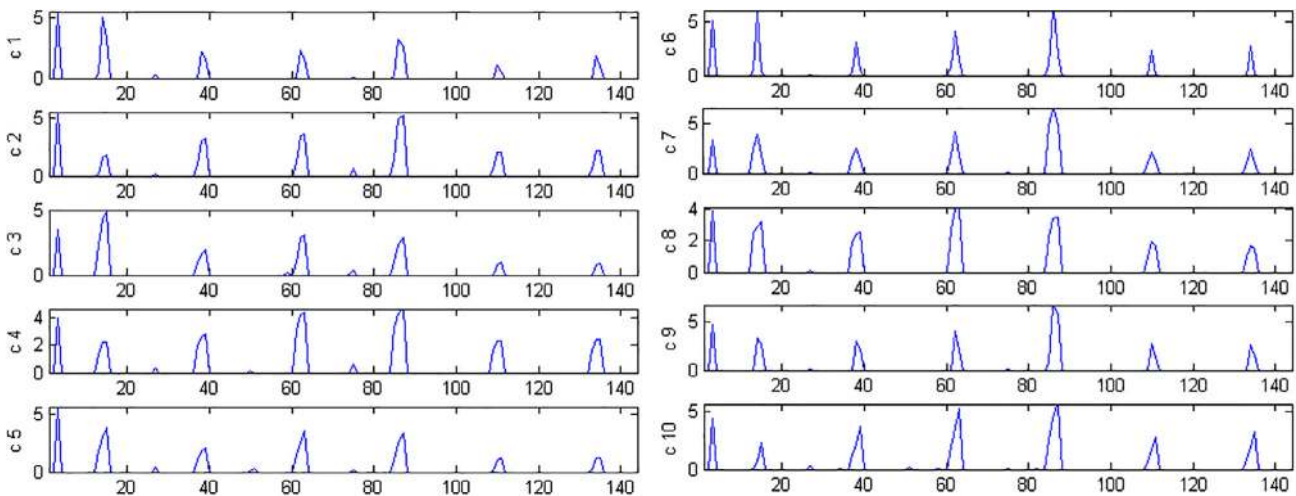


Fig. 29. Variation of CEDD features over 10 classes.

Table 14  
Percent recognition rates using texture features.

Class	Acc	Class	Acc	Class	Acc	Class	Acc	Class	Acc
63	100	64	90	65	100	66	100	67	90
68	90	69	100	70	100	71	100	72	100
73	100	74	100	75	100	76	60	77	100
78	90	79	100	80	100	81	100	82	90
83	90	84	100	85	90	86	100		

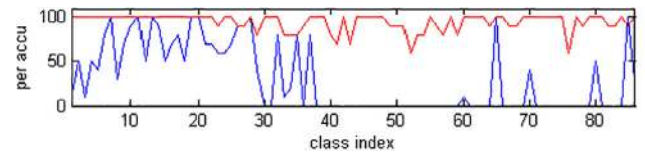


Fig. 30. Accuracy using approach (Aakif and Khan, 2015) (blue) and current approach (red).

opportunity to build the ideal opportunity for the order. General precision is 29.8%. Fig. 30 looks at exactness of the past approach (blue) with the present approach (red).

In Yang and Wang (2012) leaf shapes are displayed utilizing Fourier descriptors. For the current dataset, it was discovered that the smooth sinusoidal premise capacities were unsatisfactory for demonstrating the transient signs happening along sharp leaf shapes. General exactness is 23.02%. Fig. 31 compares the exactness of the past approach (blue) with the present approach (red).

Hierarchical clustering utilized as a part of Naresh and Nagendraswamy (2016) clusters the leaf pictures relying upon the likeness of the texture. In the event that the picture of the leaf does not contain any conspicuous structure or if the nature of the checked information is poor, at that point this strategy for clustering the pictures depending the texture reduce recognition rates.

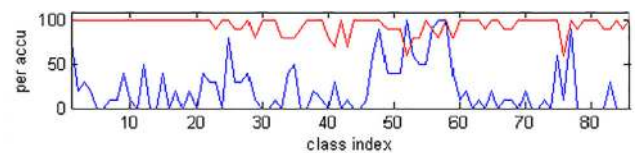


Fig. 31. Accuracy using approach (Yang and Wang, 2012) (blue) and current approach (red).

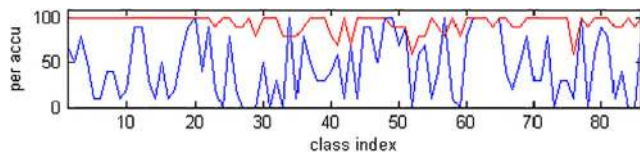


Fig. 32. Accuracy using approach (Naresh and Nagendraswamy, 2016) (blue) and current approach (red).

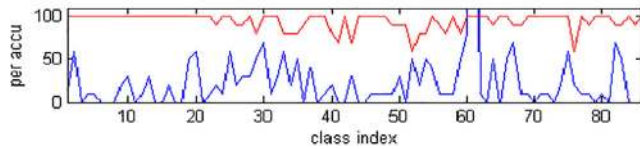


Fig. 33. Accuracy using approach (Salve et al., 2016a,b) (blue) and current approach (red).

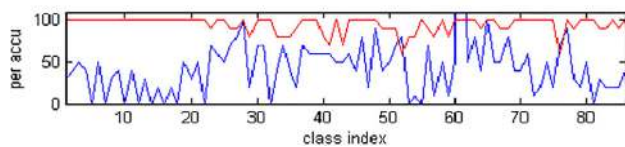


Fig. 34. Accuracy using approach (Kalyoncu and Toygar, 2015) (blue) and current approach (red).

General precision is 47.2%. Fig. 32 compares class exactness of the past approach (blue) with the present approach (red).

In Salve et al. (2016a,b) the Zernike minute and HoG technique is utilized as shape descriptor for the order of takes off. HoG depends generally on the piece estimate, cell size and number of introduction receptacles, which have a tendency to be distinctive for various shapes. General exactness is 27.4%. Fig. 33 looks at class precision of the past approach (blue) with the present approach (red).

In Kalyoncu and Toygar (2015) the Moment Invariant, convexity, perimeter ratio, multi scale distance ratio, average edge separation, and margin statistics is utilized for the order of leaves utilizing LDC. As every one of the elements are shape highlights, the leaf pictures with comparable shapes may not be taken care of by this approach. General exactness is 48.1%. Fig. 34 analyzes class exactness of the past approach (blue) with the present approach (red).

## 5. Conclusions and future scopes

This article proposes a set of techniques for using various visual attributes for classifying heterogeneous leaf sorts contrasting in hue, shape and surface. Hue based displaying is utilized for non-green leaves, shape based procedures are utilized for simple and compound leaves with clear forms, and surface based procedures are utilized for leaves with ambiguous or similar shapes. New FSST approach is utilized for the choice of shape highlights for various leaf classes. A progressive approach is taken after which comprises of a pre-processing venture, for normalizing the scale and introduction of various leaves, a hue investigation step which includes extraction of hue highlights, a shape examination step including shape based demonstrating and a surface examination venture to show surface examples of the leaf surface. Each layer comprises

of modules for custom treatment of various leaf sorts and discriminators for picking the suitable module for consequent preparing. NFC based grouping is done to exploit fuzzy likenesses between entire leaves for the hue and shape layers and Euclidean separation is utilized for segregating surface elements.

The benefits of utilizing a multi-levelled approach include the following: (1) gives an extent of consolidating numerous visual elements to produce a more total comprehension of the visual data (2) gives an extent of customization as highlight modules and classifiers can be enhanced for particular visual attributes (3) gives an extent of enhancing efficiency as discriminator modules use seek operations inside divided segments of the database (4) gives an extent of adaptability through extra layers in the system.

Future directions of research for improving the system would involve handling fragmented leaves where parts of the leaf surface areas are missing.

## References

- Aakif, M., Khan, F., 2015. Automatic classification of plants based on their leaves. *Biosyst. Eng.* 139, 66–75.
- Anjomshoae, S.T., Rahim, M.S.M., 2016. Enhancement of template-based method for overlapping rubber tree leaf identification. *Comput. Electron. Agric.* 122, 176–184.
- Caoa, J., Wanga, B., Brown, D., 2016. Similarity based leaf image retrieval using multiscale R-angle description. *Inf. Sci.* 374, 51–64.
- Chaki, J., Parekh, R., Bhattacharya, S., 2015. Plant Leaf Recognition using Texture and Shape features with Neural Classifiers. *Pattern Recognit. Lett.* 58, 61–68.
- Plantscan Dataset ([http://imedia-ftp.inria.fr:50012/Pl@ntNet/plantscan\\_v2/](http://imedia-ftp.inria.fr:50012/Pl@ntNet/plantscan_v2/)).
- De Souzaa, M.M.S., Medeiros, F.N.S., Ramalho, G.L.B., de Paula Jr, I.C., Oliveira, I.N.S., 2016. Evolutionary optimization of a multiscale descriptor for leaf shape analysis. *Expert Syst. Appl.* 63, 375–385.
- Deng, W., Huang, Y., Zhao, C., Chen, L., Wang, X., 2016. Bayesian discriminant analysis of plant leaf hyperspectral reflectance for identification of weeds from cabbages. *Afr. J. Agric. Res.* 11, 551–562.
- Fotopoulou, F., Laskaris, N., Economou, G., Fotopoulos, S., 2013. Advanced leaf image retrieval via Multidimensional Embedding Sequence Similarity (MESS) method. *Pattern Anal. Appl.* 16, 381–392.
- Kadir, A. et al., 2012. Experiments of Zernike moments for leaf identifications. *Theor. Appl. Inf. Technol.* 41, 82–93.
- Kalyoncu, C., Toygar, O., 2015. Geometric leaf classification. *Comput. Vision Image Understanding* 133, 102–109.
- Kumar, N. et al., 2012. Leafsnap: a computer vision system for automatic plant species identification. *LNCS 7573*, 502–516.
- Le, T.L., Tran, D.T., Pham, N.H., 2014. Kernel Descriptor Based Plant Leaf Identification. *IEEE Image Processing Theory, Tools and Application*.
- Liu, N., Kan, J.M., 2016. Plant leaf identification based on the multi-feature fusion and deep belief networks method. *J. Beijing Univ.* 8, 110–119.
- Naresh, Y.G., Nagendraswamy, H.S., 2016. Classification of medicinal plants: an approach using modified LBP with symbolic representation. *Neurocomputing* 173, 1789–1797.
- Prasad, S., Kumar, P.S., Ghosh, D., 2016. An efficient low vision plant leaf shape identification system for smart phones. *Multimedia Tools Appl.*, 1–25.
- Sakai, N., Yonekawa, S., Matsuzaki, A., 1996. Two dimensional image analysis of the shape of rice and its applications to separating varieties. *J. Food Eng.* 27, 397–407.
- Salve, P., Sardesai, M., Manza, R., Yannawar, P., 2016a. Identification of the plants based on leaf shape descriptors. *Springer Adv. Intell. Syst. Comput.* 379, 85–101.
- Salve, P., Sardesai, M., Manza, R., Yannawar, P., 2016b. Identification of the plants based on leaf shape descriptor. *Springer AISC 376*, 85–101.
- Scharr, H., Minervini, M., French, A.P., Klukas, C., Kramer, D.M., Liu, X., Luengo, I., Pape, J.M., Polder, G., Vukadinovic, D., Yin, X., Tsafaris, S.A., 2016. Leaf segmentation in plant phenotyping: a collation study. *Mach. Vision Appl.* 27, 585–606.
- Flavia Plant Leaf Recognition System (<http://sourceforge.net/projects/flavia/files/Leaf%20Image%20Dataset/>).
- Wang, Q.P., Du, J.X., Zhai, C.M., 2010. Recognition of leaf image based on ring projection wavelet fractal feature. *LNCS 6216*, 240–246.
- Wang, Z.C., Feng, D., 2002. Fuzzy integral for leaf image retrieval. *IEEE Int. Conf. Fuzzy Syst.*, 372–377.
- Wang, Z., Sun, X., Ma, Y., Zhang, H., Ma, Y., Xie, W., Zhang, Y., 2014. Plant recognition based on intersecting cortical model. *IEEE IJCNN*, 975–980.
- Yang, L.F., Wang, X.F., 2012. Leaf image recognition using Fourier Transform based on ordered sequence. *Springer LNCS 7389*, 393–400.
- Zhang, S., Lei, Y., Zhang, C., et al., 2016. Semi-supervised orthogonal discriminant projection for plant leaf classification. *Pattern Anal. Appl.* 19, 953–961.