# Computer-aided Interpretable Features for Leaf Image Classification

**Abstract**

Plant species identification is time consuming, costly, and requires lots of efforts, and expertise knowledge. The studies on medicinal plant identification are often performed based on medinal plant leaf images. The is because plant leaves contain a large number of diverse set of features such as shape, veins, edge features, apices etc. that are useful in identifying different medicinal plants. In recent, many researchers use deep learning methods to classify plants directly using plant images. While deep learning models have achieved a great sucess, the lack of interpretability limit their widespread application. To overcome this, we explore the use of interpretable, measurable and computer-aided features extracted from plant leaf images. Image processing is one of the most challenging, and crucial steps in feature-extaction. The purpose of image processing is to improve the leaf image by removing undesired distortion. The main image processing steps of our algorithm involves: i) Convert original image to RGB (Red-Green-Blue) image, ii) Gray scaling, iii) Gaussian smoothing, iv) Binary thresholding, v) Remove stalk, vi) Closing holes, and vii) Resize image. The next step after image processing is to extract features from plant leaf images. We introduced 52 computationally efficient features to classify plant species. These features are mainly classified into four groups as: i) shape-based features, ii) color-based features, iii) texture-based features, and iv) scagnostic features. Length, width, area, texture correlation, monotonocity and scagnostics are to name few of them. We explore the ability of features to discriminate the classes of interest under supervised learning and unsupervised learning settings. For that, supervised dimensionality reduction techique Linear Discriminant Analysis (LDA), and unsupervised dimensionality reduction techique as Principal Component Analysis (PCA) are used to convert and visualize the images from digital-image space to feature space. All the applications are performed on Flavia and Swedish open source benchmark leaf datasets. The results show that the features are sufficient to discriminate the classes of interest under both supervised and unsupervised learning settings. The results of this study are beneficial for the researchers working in the field of developing automated plant identification and classification systems.

## 1 Introduction

Leaf identification is becoming very popular in classifying plant species. Plant leaf contains significant number of features that can help people to identify and classify the plant species. In medical perspective, medicinal plants are usually identified by practitioners based on years of experience through sensory or olfactory senses. The other method of recognizing these plants involves laboratory-based testing, which requires trained skills, data interpretation which is costly and time-intensive. Automatic ways to identify medicinal plants are useful especially those that are lacking experience in medicinal plant recognition. Statistical machine learning techniques play a crucial role in the development of automatic system to identify medicinal plants. In developing such system image processing and feature extraction play crucial roles.

The main aim of image processing is to extract important features by removing undesired noise and distortion (Waldchen and Mader 2018). Image processing steps include image segmentation (Anantrasirichai, Hannuna, and Canagarajah 2017), image orientation, cropping, grey scaling, binary thresholding, noise removal, contrast stretching, threshold inversion, image normalization, and edge recognition are some of image processing techniques applied in recent research. These steps can be applied parallely or individually, on several times until the quality of leaf image reaches a specific threshold.

The second step is feature extraction, which identifies and encodes relevant features from leaf images (Waldchen and Mader 2018). This is a challenging task due to the structural diversity of the leaf images.

Therefore, in recent years, many researchers use deep learning methods to classify plants directly using plant images (Wu et al. 2007; Azlah et al. 2019; Herdiyeni and Wahyuni 2012). While deep learning models have achieved a great sucess, the most models remain complex black boxes. The lack of interpretability limit their widespread application.

A digital image is a combination of pixels from three different color planes red, green and blue. Images are stored in computers as three separate matrices corresponding to the red, green and blue color channels of the image. The three separate matrices corresponding to intensities of colors at different positions of the image (Gonzalez and Woods 2006).

The main aim of feature extraction is to reduce dimensionality of this information by obtaining measurable patterns of leaf images. For example, shape, color, and texture are some of the patterns that may be observed. In this paper we introduce a collection of interpretable, measurable and computer-aided features that are useful in image leaf classification. This feature collection includes several pre-established features identified through a thorough review of literature. Other than existing features, we introduce a number of features computed based on the cartesian coordinate of the images. Furthermore, we explore the ability of features to discriminate the class of interest under supervised learning setting and unsupervised learning setting.

The paper is organized as follows. In the first section, describes about the steps of image processing. Preprocessing is necessary before extracting features from the images. In section 2, discusses about feature extraction in in-detail because features are highly influenced by the plant species to be classified. Under this section, we discuss about mainly four types of features as shape, color, texture, and scagnostics and how to extract them. The next section 3, discuss about applications of leaf features. This section consists of details about the datasets that are used to explain the applications, and visualization of leaf images in the feature space using supervised, and unsupervised dimensionality reduction techniques. In section 4, consist of summary of the software and packages that use to extract the features. Some discussion about the outputs and concluding remarks are given in last section.

## 2 Image Processing

Image processing is an essential step to reduce noise and content enhancement while keeping its features intact. (Goyal, Kapil, and Kumar 2018). The workflow we use to process images in this paper is shown in Figure 1. This includes seven main steps. They are: i) converting BGR (Blue-Green-Red) image to RGB (Red-Green-Blue), ii) gray scaling, iii) Gaussian filtering, iv) binary thresholding, v) remove stalk, vi) close holes, and vii) image resizing. Some of these steps are applicable only for sepecific images. For example, apply remove stalk is applicable only to leaf images which has stalk.

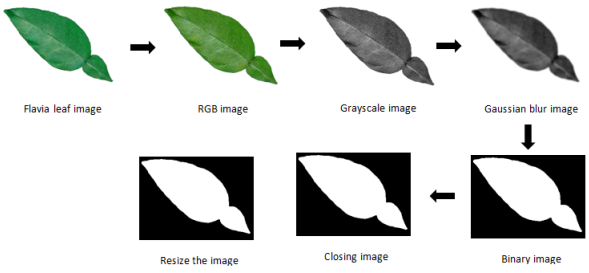



Figure 1: Image processing workflow.

In this study we focus on leaves with simple arrangement as shown in Figure 2. A single leaf that is never divided into smaller leaflet units is known as a leaf with simple arrangement. This type of leaf is attached to a twing by its stem or the petiole. The margins, or edges, of the leaf can be smooth, lobed, or toothed (see Figure 3).

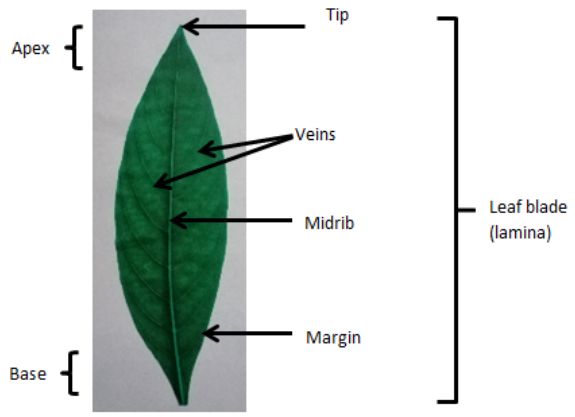



Figure 2: A leaf with simple arrangement.

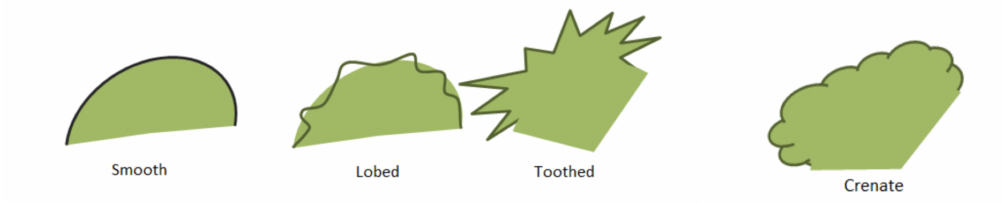



Figure 3: Edge types of leaves focus in the study.

## 2.1 Step 1: Converting BGR (Blue-Green-Red) Image to RGB (Red-Green-Blue)

BGR and RGB are conventions for the order of the different colour channels. They are not colour spaces. When converting BGR image to RGB, there is no any computations, just switches around the order. Several image processing libraries have different pixel ordering. Therefore to compatible with other libraries we convert the BGR image in to RGB format. For an example when we read an image using OpenCV libray in Python by default it interprets BGR format, but when we plot the image it takes the RGB format in matplotlib package in Python.

## 2.2 Step 2: Grayscaling

Grayscaling is the process of converting an image to shades of gray from other colour spaces like RGB. This helps to increase the contrast and intensity of images (Goyal, Kapil, and Kumar 2018). Grayscale images require only one single byte for each pixel where as colour (RGB) image requires 3 bytes for each pixel. Hence, grayscaling reduces the dimension of a image.

In extracting Haralick (Haralick, Shanmugam, and Dinstein 1973) texture features, grayscale images are used. Another advantage of using grayscaled image is to reduce model complexity. Consider an example of training neural article on RGB images of $20 \times 20 \times 3$ pixel. The input layer will have 1200 input nodes. Whereas for gray scaled images the same neural network will need only 400 ($20 \times 20$) input node.

## 2.3 Step 3: Gaussian Filtering (Gaussian Blurring/Gaussian Smoothing)

Gaussian smoothing is an image smoothing technique. Image smoothing techniques use to remove the noise that can be occurred because of the source (camera sensor). Image smoothing techniques help in smoothing images and to remove low intensity edges.

Gaussian function is used to blur the image. It is a linear filter which is done by using the functions in OpenCV package in Python. By specifying the width and height of the Gaussian kernel that must be positive and odd, and specifying the kernel standard deviation along x and y-axis, Gaussian smoothing is established in OpenCV. When the kernel standard deviation along x-axis is specified, kernel standard deviation along y-axis is taken as equal to the the kernel standard deviation along x-axis. But if both kernel standard deviation are given as zeros, they are calculated by using the kernel size. In our research the width and height of the kernel is defined as 55 and the kernel standard deviation along x-axis is assigned as zero. An example of applying Gaussian smoothing to an image is shown in Figure **??**.

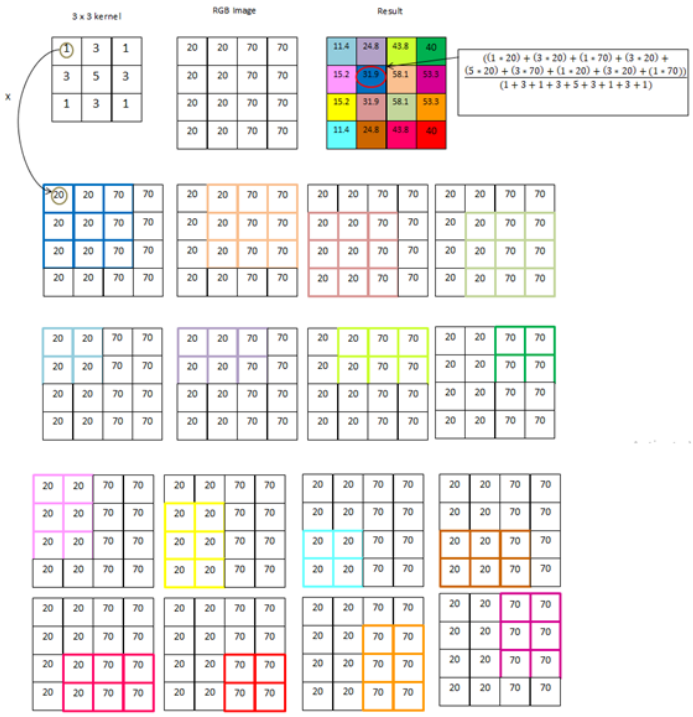



Figure 4: Example of gaussian smoothing

## 2.4 Step 4: Binary Thresholding

Thresholding is a segmentation technique that is used to separate foreground from its background. Thresholding converts

grayscale image into binary image according the threshold value. If the pixel value is smaller than the threshold value, the pixel value is set as 0, and if not the pixel value is set to a maximum value which is generally 255. Thersholding technique is done on grayscale images in computer vision.

We used Otsu's binarization (Bangare et al. (2015)) which is an adaptive thresholding after Gaussian filtering to convert color images to the binary images. The reason for using Otsu's binarization method is, it automatically determines the optimal threshold value.

**Otsu's Thresholding**

Nobuyuki Otsu introduced Otsu's method which is defined for a histogram of grayscale values of a histogram ($ghist_I$) of an input image $Im$. To segment an image $Im$ into two subsets of pixels Otsu's method calculates an optimal threshold $\tau$. The number of pixel locations of the gray scale image is defined as $|\Omega|$.

The algorithm maximizes the variance $\sigma^2$ between the two subsets (Within-class-variance) to find the threshold $\tau$.

$$\sigma^2 = P_1(\mu_1 - \mu)^2 + P_2(\mu_2 - \mu)^2 = P_1 P_2(\mu_1 - \mu_2)^2$$

where $\mu$ is the mean of the histogram, $\mu_1$ and $\mu_2$ are the mean values of first and second subset respectively, $P_1$ and $P_2$ are the corresponding probabilities of the two clusters, defined by

$$P_1 = \frac{\sum_{i=0}^{u} ghist_I(i)}{|\Omega|}$$

$$P_2 = \frac{\sum_{i=u+1}^{255} ghist_I(i)}{|\Omega|}$$

Where $u$ is the candidate threshold and the maximum gray level ($G_{max}$) is assumed as 255. To find optimal threshold $\tau$ for segmenting image $Im$, all candidate thresholds are evaluated this way.

The algorithm of Otsu's method is defined as follows,

| |
|---|
| Create a histogram for the grayscale image |
| Set the histogram variance $S_{max} = 0$ |
| **while** $u < G_{max}$ **do** |
| Compute $\sigma^2 = P_1 * P_2(\mu_1 - \mu_2)^2$ |
| **if** $\sigma^2 > S_{max}$ **then** |
| $S_{max} = \sigma^2$ |
| $\tau = u$ |
| **end if** |
| Set $u = u + 1$ |
| **end while** |

Table 1: Otsu's method

For an example, assume that candidate threshold value $u$ is 2. Therefore the image is separated into two classes, which are class 1 (pixel value <= 2 ) and class 2 (pixel value > 2). Class 1 represents the background and class 2 represents the foreground of the grayscale image. According to figure 5, there are 9 pixel locations.

$$P_1 = \frac{5}{9}$$

$$P_1 = \frac{4}{9}$$

$$\mu_1 = \frac{(0*2) + (1*1) + (2*2)}{(2+1+2)} = 1$$

$$\mu_1 = \frac{(3*3) + (4*1)}{(3+1)} = \frac{13}{4}$$

$$\sigma^2 = \frac{5}{9} * \frac{4}{9} * (1 - \frac{13}{4})^2 = 1.25$$

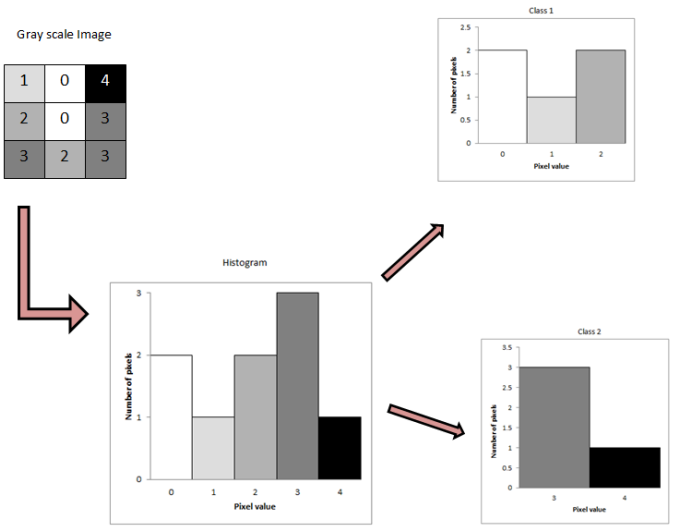



Figure 5: Example of Otsu's binary thresholding

## 2.5 Step 5: Image Resizing

In this study we use two different benchmark leaf image datasets: i) Flavia: 1907 fully color images of 32 classes of leaves (Wu et al. (2007)) and ii) Swedish: 1125 images from 15 different plant species (Söderkvist (2001)). Flavia and Swedish images have two different image sizes. To compare the results on different datasets, to improve the memory storage capacity and to reduce computational complexity the leaf images are resized to a fixed resolution. In our study, the leaf images have been resized to [1600 x 1200px] which is the size of Flavia leaf images.

Other than the main image processing techniques discussed above, the following techniques are applied to some images where necessary after image thresholding.

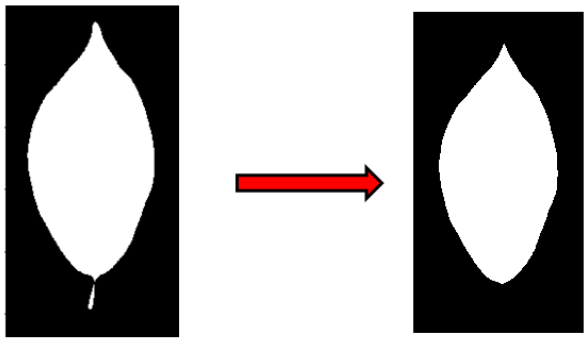


Figure 6: Remove stalk

### 2.5.1 Remove Stalk

As shown in Figure 6, this is used to remove the stalk of the image. Remove the petiole (stalk) of leaf image is another version of thresholding process. Thresholding is applied after finding the sure foreground area. To find the sure foreground area, distance transform technique is used. Binary image is used as the input of distance transform technique. In distance transform technique, image is created by assigning a

number for each object pixel that corresponds to the distance to the nearest background pixel. The distance is calculated using the euclidean distance (see Figure 7). After finding the sure foreground area, Otsu's binarization is applied again as the thresholding technique.

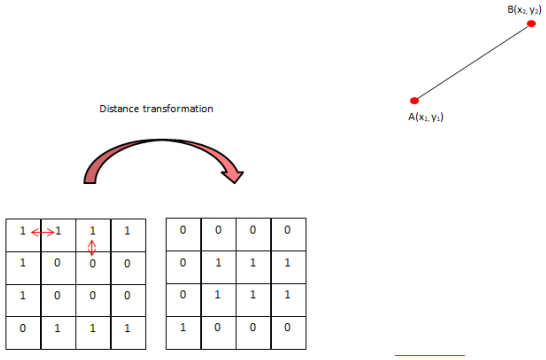



Figure 7: Example of distance transformation

$$\text{Euclidean distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

### 2.5.2 Closing Holes

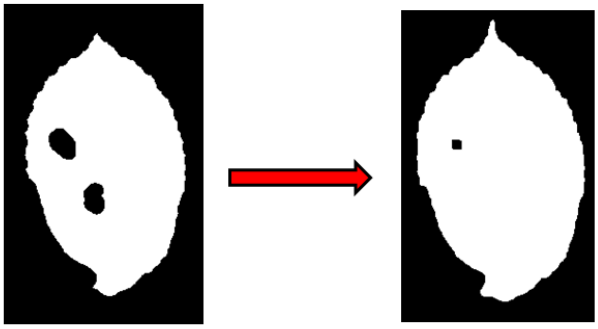


Figure 8: Closing holes

Holes inside leaf areas occure due to plant disease, light reflection and noise. As shown in 8 closing holes is used to remove small holes inside the foreground objects. This is achieved by pulling background pixels to foreground pixel. The closing holes is also known as dilation and while erosion does the opposite, which is open holes. This step is performed on a binary image.

**Erosion**

The basic idea of Erosion is that erodes away the boundaries of foreground object. Since the input is binary image, a pixels in the original image is either 1 or 0. If all the pixels under the kernel is 1, a pixel of original image is considered as 1, otherwise made to zero (eroded). Which means that depending upon the size of the kernel all pixels near boundary will be discarded. Therefore the thickness or size of the foreground object decreases (White region of the image decreases).

**Dilation**

Opposite of erosion is defined as dilation. If at least one pixel under the kernel is 1, the pixel element is 1 in Dilation. It tends to increase the foreground of the image or the white region of the object.

# 3 Leaf Image Features

In identification of plant species using leaf images, features of leaves play an important role, because each leaf posses unique feature that it make different from other. In previous studies (Azlah et al. 2019; Jeon and Rhee 2017; Sun et al. 2017), use deep learning neural networks to classify medicinal plants based on pixel-based images. Given the leaf images deep learning models automatically identify features based on pixel-space of the images. While deep learning models have achieved a great sucess, the lack of interpretability of features limit their widespread application. To overcome this, we explore the use of interpretable, measurable and computer-aided features extracted from plant leaf images. We identified 52 features. The features are classified into four groups as: i) shape-based features, ii) color-based features, iii) texture-based features, and iv) scagnostic features.

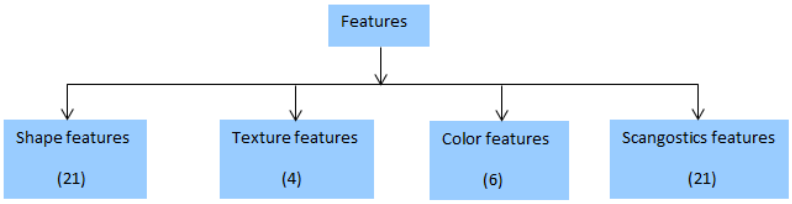



Figure 9: Classification of features and their composition

## 3.1 Shape Features

When identifying real-world objects, the shape is known as an essential sign for humans. We use the shape descriptors introduced by Waldchen and Mader (2018). AInaddition to that we introduce a number of new shape features such as: number of convex points, x and y coordinates of center, number of maximum and minimum points, correlation of cartesian contour points, etc. The shape features should be invariant to certain class of geometric transformation of the object. The main geometric transformation are rotation reflection, scaling and translation (see figure 10). The shape features we considered in this study are invariant to the rotation and reflection. All the shape features are extracted from the binary images.

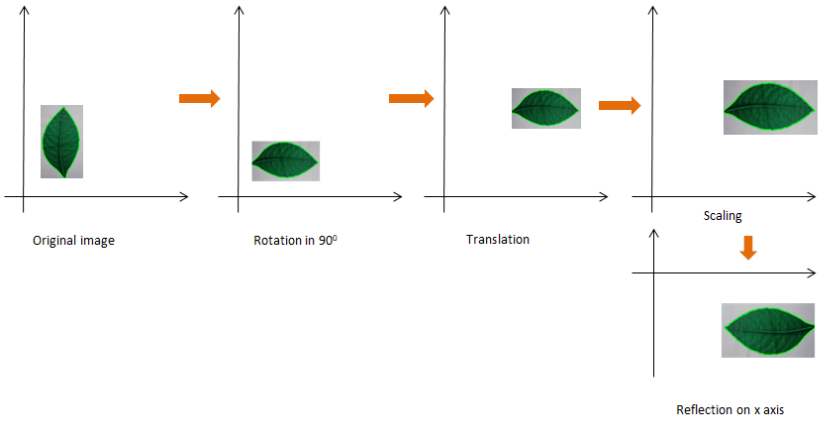



Figure 10: Illustration of geometric transformation

Extraction of image contour plays an important role in measuring the shape of a image. Simply contour

(see figure 11) is a curve joining all the continuous points (along the boundary), having the same color or intensity.

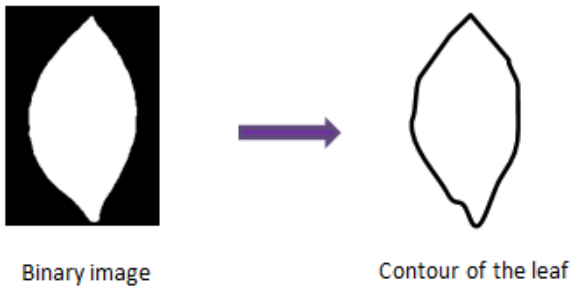



Figure 11: Extract contour of the leaf image

In order to extract the contour of the leaf, the leaf should be placed properly in the center of a white paper. If the image is place as shown in Figure 12, as a result of inappropriate translation (a) and inappropriate scaling (b), problems arises in the calculations of the contour. Furthermore, it is difficult to recognize the contour, when the image is too small.

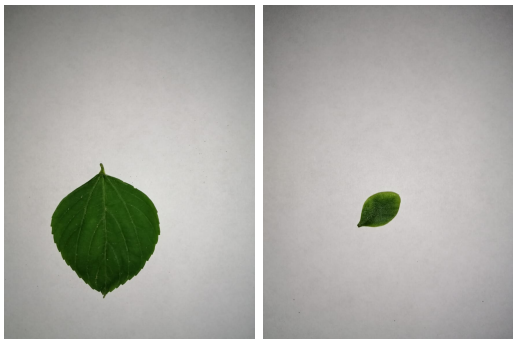



Figure 12: Inappropriate translation (a), Inappropriate scaling (b

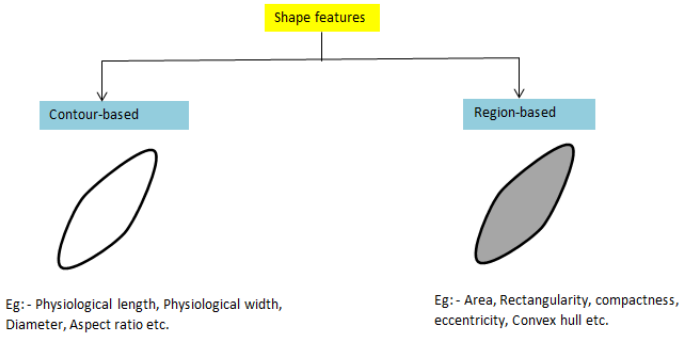



Figure 13: Categorization of shape features

Shape features can be classified into two main categories as contour-based and region-based features (Waldchen and Mader 2018). As illustrated in Waldchen and Mader (2018), contour-based shape features are

computed based on the contour of a shape, whereas region-based shape features are extracted from the whole region of a shape (see figure 13).

In this study, we use 6 initial shape features that are used to derive 12 shape features. The 6 initial features are: i) diameter, ii) physiological length, iii) physiological width, iv) area, v) perimeter, and vi) eccentricity.

### 3.1.1 Diameter ($F_1$)

Diameter is defined as the longest distance between any two points on the margin of the leaf (Waldchen and Mader 2018). To calculate the diameter of the leaf image, first we need to find the contour of the leaf image. Then we need to select all pair of contour points and measure the Euclidean distance (equation 2) between the two points separately. Finally have to find the maximum distance among the calculated distances. (see figure 14)

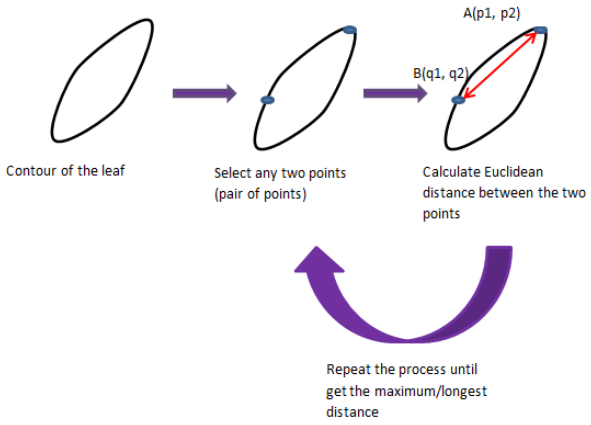



Figure 14: Logic behind calculation of diameter

$$d\left(A,B\right)=\sqrt{\sum_{i=1}^{2}\left(q_i-p_i\right)^2} \tag{2}$$

$$F_1 = max(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}); \forall i, j, i \neq j \tag{3}$$

### 3.1.2 Physiological length ($F_2$) and Physiological width ($F_3$)

According to the authors in Azlah et al. (2019), the physiological length is "measured based on the main vein of leaf, as it stretches from the main vein to the end tip". We use the definition by Azlah et al. (2019), the physiological width is "the span of leaf viewed from one side to the other, from the leftmost point to the rightmost point of leaf" . There are straight (horizontal or vertical) and angled leaf images in our datasets, Flavia and Swedish (see example Figure 15). There are two types of bounding rectangles.

i) Straight bounding rectangle: This is a straight rectangle which does not consider the rotation of the object (see Figure 15 (a)).

ii) Rotated rectangle: This bounding rectangle is drawn with minimum background area. Therefore the rotation of the object is also considered ((see Figure 15 (b)).

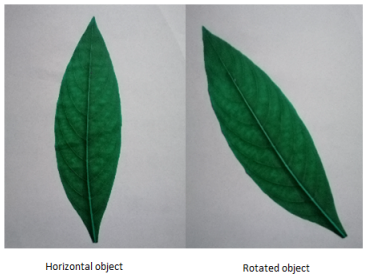
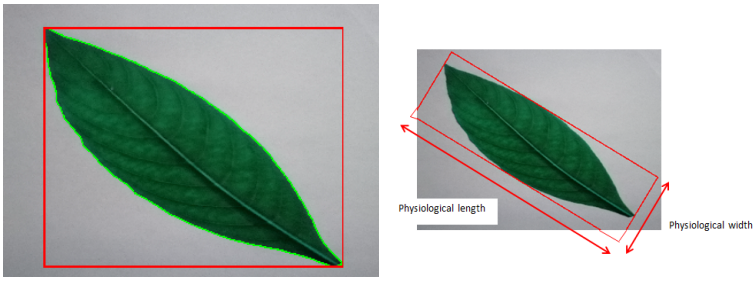



Figure 15: Straight (horizontal or vertical) and rotated leaf image in Actual leaf image dataset

The straight bounding rectangle is enough to extract physiological length and physiological width of straight (horizontal or vertical) leaf images. However, the straight bounding rectangle is not suitable to compute physiological length and physiological width of angled leaf images. To solve this problem, we considered rotated rectangle rather than bounded rectangle in computing shape features of angled images. As shown in Figure **??**, physiological length and width are computed as follows:

$$F_2 = \text{Length of the rotated rectangle} \tag{4}$$

$$F_3 = \text{Width of the rotated rectangle} \tag{5}$$




(a) (b)

Figure 16: Straight bounded rectangle of a rotated leaf image (a), Rotated rectangle of angled leaf image (b)

### 3.1.3 Area ($F_4$)

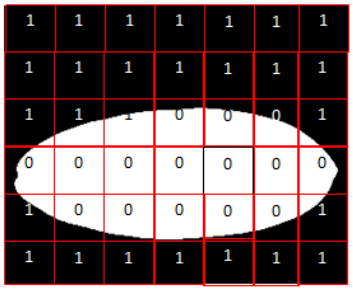



Figure 17: Measure the area

$$F_4 = \text{Number of zero pixels covered by the contour} \tag{6}$$

Area is computed after applying the thresholding process. Next, we need to extract the best contour and based on that contour area is measured. Number of 0 pixels covered by the contour is the measure of area of leaf image (Azlah et al. 2019). As shown in Figure 18 the area measures increase as the area of the leaf increases.

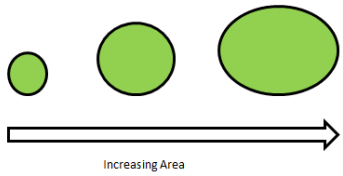



Figure 18: Area

### 3.1.4 Perimeter ($F_5$)

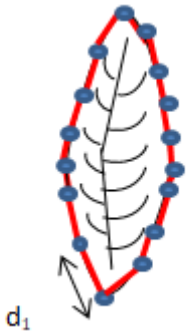



Figure 19: Area

As shown in Figure 19 perimeter is defined as the summation of euclidean distance of all continuous points in the contour. Let $n$ be the number of distances around the contour, then the perimeter is defined as

$$F_5 = \sum_{i=0}^{n} d_i \tag{7}$$

### 3.1.5 Eccentricity ($F_6$)

Eccentricity is a characteristic of any conic section of a leaf (Azlah et al. 2019). Eccentricity is defined that how much the ellipse actually varying being circular. Eccentricity is calculated using equation 8.

$$F_6 = \sqrt{1 - \frac{b^2}{a^2}} \tag{8}$$

where $a$ is semi-major axis and $b$ is semi-minor axis (see Figure 20)

Eccentricity of an ellipse is varied between 0 and 1. If eccentricity is 0, then we obtain a circle whereas eccentricity is 1, then we obtain an ellipse (see Figure 21).

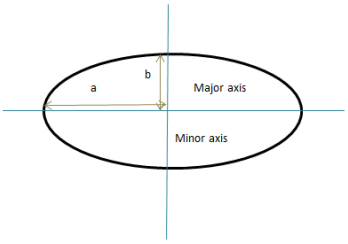



Figure 20: Ellipse

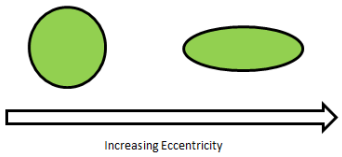



Figure 21: Eccentricity

### 3.1.6 Number of Convex Points ($F_{21}$) and Verticies

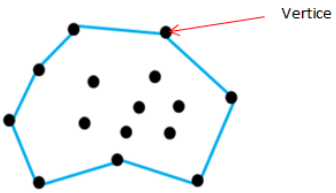



Figure 22: Convex hull

Wilkinson, Anand, and Grossman (2005) defined a convex hull as a "hull that contains all the straight line segments connecting any pair of points in its interior". The convex hull bounds a single polygon (see Figure 22). We introduced two new features computed based on the convex hull: i) Number of convex points ($F_2$1) and ii) Number of convex vertices ($F_7$).

### 3.1.7 Roundness/ Circularity

Roundness is named as aka form factor, circularity, or isoperimetric factor. Roundness illustrates the difference between the leaf and a circle. Equation 9 is used to calculate roundness. Figure 23 shows what happens to the shape when the roundness measure is changed.

$$R = \frac{4\pi F_4}{{F_5}^2} \tag{9}$$

### 3.1.8 Compactness

Compactness is closely related to roundness. Compactness measures that how compatible the leaf fits to a circle area (see figure 24).

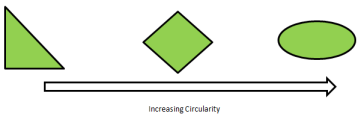



Figure 23: Circularity

$$C = \frac{{F_5}^2}{F_4} \tag{10}$$

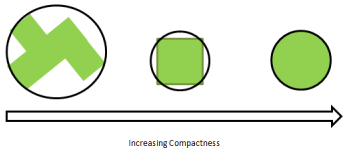



Figure 24: Compactness

### 3.1.9 Convexity

Convexity measures the curvature of the convex hull.

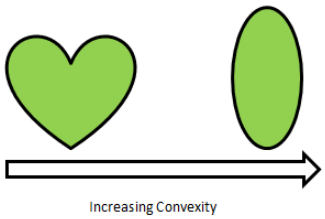



Figure 25: Convexity

The equations and sfotware packages used to compute shape features are shown in Table 2.

| Shape feature | Feature name | Figure | Formula | Range | Software package |
|---|---|---|---|---|---|
| $F_1$ | Diameter |  | eq:3 | [0,∞] | combinations, numpy |
| $F_2$ | Physiological length |  | eq:4 | [0,∞) | OpenCV |
| $F_3$ | Physiological width |  | eq:5 | [0,∞) | OpenCV |
| $F_4$ | Area |  | eq:6 | | OpenCV |

**Table 2 continued from previous page**

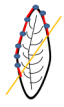
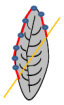
| Shape feature | Feature name | Figure | Formula | Range | Software package |
|---|---|---|---|---|---|
| $F_5$ | Perimeter |  | eq:7 | [0,∞) | OpenCV |
| $F_6$ | Eccentricity | | eq:8 | [0,1] | OpenCV |
| $F_7$, $F_8$ | x and y coordinate of center |  | | | scipy.ndimage |
| $F_9$ | Aspect ratio |  | $F_9 = \frac{F_2}{F_3}$ | [0,∞) | - |
| $F_{10}$ | Roundness/ Circularity |  | eq:9 | [0,∞) | numpy |
| $F_{11}$ | Compactness |  | eg:10 | (0,∞) | - |
| $F_{12}$ | Rectangularity | | $F_{12} = \frac{{F_5}^2}{F_4}$ | (0,∞) | - |
| $F_{13}$ | Narrow factor |  | $F_{13} = \frac{F_1}{F_2}$ | [0,∞) | - |
| $F_{14}$ | Perimeter ratio of diameter |  | $F_{14} = \frac{F_5}{F_1}$ | [0,∞) | - |
| $F_{15}$ | Perimeter ratio of physiological length |  | $F_{15} = \frac{F_5}{F_2}$ | [0,∞) | - |
| $F_{16}$ | Perimeter ratio of physiological length and width |  | $F_{16} = \frac{F_5}{F_2 * F_3}$ | [0,∞) | - |
| $F_{17}$ | Perimeter convexity |  | $F_{17} = \frac{\text{Perimeter of convex hull}}{F_5}$ | [0,∞) | OpenCV |
| $F_{18}$ | Area convexity |  | $F_{18} = \frac{(\text{Area of convex hull} - F_4)}{F_4}$ | [0,∞) | OpenCV |
| $F_{19}$ | Area ratio of convexity |  | $F_{19} = \frac{F_4}{\text{Area of convex hull}}$ | [0,∞) | OpenCV |

**Table 2 continued from previous page**

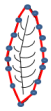
| Shape feature | Feature name | Figure | Formula | Range | Software package |
|---|---|---|---|---|---|
| $F_{20}$ | Equivalent diameter |  | $F_{20} = \sqrt{\frac{4*F_4}{\pi}}$ | [0,∞) | numpy |
| $F_{21}$ | Number of convex points |  | number of convex points | [0,∞) | OpenCV |

Table 2: Summary of shape features.

## 3.2 Texture Features

Texture features are used to describe the surface or the appearance of the leaf image. Texture can be assessed using a group of pixels. Color is a property of a pixel. Texture is defined as feel of various materials to human touch and texture is quantified based on visual interpretation of this feeling. Leaf surface is a natural texture which has random persistent patterns and do not show detectable quasi-periodic structure (Waldchen and Mader 2018). Therefore to describe the natural texture patterns of the leaf fractal theory (Waldchen and Mader 2018) is the best approach.

The Haralick texture features (Boland 2000) are functions of the normalized GLCM (Gray Level Co-occurrence Matrix) which is a common method to represent image texture.

$$GLCM = \begin{bmatrix} h(1,1) & h(1,2) & \cdot & \cdot & \cdot & h(1,n) \\ h(2,1) & h(2,2) & \cdot & \cdot & \cdot & h(2,n) \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ h(n,1) & h(n,2) & \cdot & \cdot & \cdot & h(n,n) \end{bmatrix}$$

The GLCM is square with dimension $n$, where $n$ is the number of gray levels in the image. Let $h(a,b)$ is the probability that a pixel with value $a$ will be found adjacent to a pixel of value $b$. Then $h(a,b)$ is defined as

$$h(a,b) = \frac{\text{Number of times a pixel with value a is adjacent to a pixel with value b}}{\text{Total number of such comparisons made}}$$

In order to calculate $h(a,b)$, adjacency can be defined to occur in each of four directions in a 2D (see figure 28), square pixel image (horizontal, vertical, left and right diagonals - see equation 26).

The Haralick statistics (Boland 2000) are calculated based on the matrices generated using each of these directions (see figure 27) of adjacency. Haralick introduced 14 statistics to describe the texture of the image based on the four co-occurrences matrices generated. In this research, we only used the following 4 statistics among 14 of them, because most of the researchers used these 4 statistics as texture features (see figure 3) of leaf images. All the texture features are extracted from the gray scale image. Texture features are calculated from the mahotas package in Python. Table 3 shows the definitions of texture features.

where h(a, b) = Probability density function of gray - level pairs $(a,b)$ and dimension of GLCM is $n * n$ (Number of columns $*$ Number of rows)

$$\mu_x = \sum_{a=1}^{columns} a \sum_{b=1}^{rows} h(a,b)$$

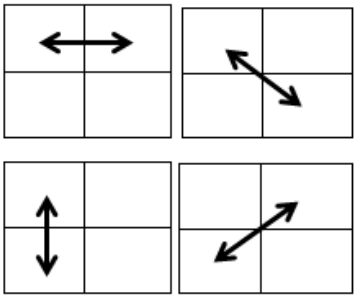


Figure 26: Four directions of adjacency as defined for calculation of the Haralick texture features

| Texture feature | Feature name | Detailed description | Formula | Value range |
|---|---|---|---|---|
| $F_{22}$ | Contrast | Measures the relation or difference between the highest and lowest gray levels of the GLCM | $\frac{\sum_{a=1}^{columns}\sum_{b=1}^{rows}(a-b)^2 h(a,b)}{\text{Number of gray levels}-1}$ | [0,∞] |
| $F_{23}$ | Entropy | Measures the randomness which means that how uniform the image is | $-\sum_{a=1}^{columns}\sum_{b=1}^{rows} h(a,b)log_2(h(a,b))$ | [−∞,0] |
| $F_{24}$ | Correlation | Measurement of dependence of gray levels of the GLCM. It measures that how a particular pixel is correlated to it's neighbor pixel over the whole image | $\frac{\sum_{a=1}^{columns}\sum_{b=1}^{rows}(ab)h(a,b)-\mu_x\mu_y}{\sigma_x\sigma_y}$ | [-1,1] |
| $F_{25}$ | Inverse difference moments | It's a measure of homogeneity. Inverse level of contrast that measures how close the values of GLCM to diagonal values in GLCM | $\sum_{a=1}^{columns}\sum_{b=1}^{rows}\frac{h(a,b)}{(a-b)^2}$ | [0,∞] |

Table 3: Definitions of texture features

$$\mu_y = \sum_{b=1}^{rows} b \sum_{a=1}^{columns} h(a,b)$$

$$\sigma_x = \sum_{a=1}^{columns} (a-\mu_x)^2 \sum_{b=1}^{rows} h(a,b)$$

$$\sigma_y = \sum_{b=1}^{rows} (b-\mu_y)^2 \sum_{a=1}^{columns} h(a,b)$$

## 3.3 Color Features

Color is an important feature of images (Waldchen and Mader 2018; Caglayan, Guclu, and Can 2013). Color properties are defined within a particular color space like red-green-blue (RGB) (Kodituwakku and S.Selvarajah 2010; Waldchen and Mader 2018). Color properties can be extracted from images after a colour space is specified. In the field of image recognition, a number of general color descriptors have been introduced. Color moments (Kodituwakku and S.Selvarajah 2010; Waldchen and Mader 2018) are the simple descriptor among them. Mean, standard deviation skewness and kurtosis are the common moments. Color moments are used for characterizing planar color patterns, irrespective of viewpoint or illumination conditions and without the need for object contour detection (Waldchen and Mader 2018). Color moments are convenient for real-time applications because of its low dimension and low computational complexity.

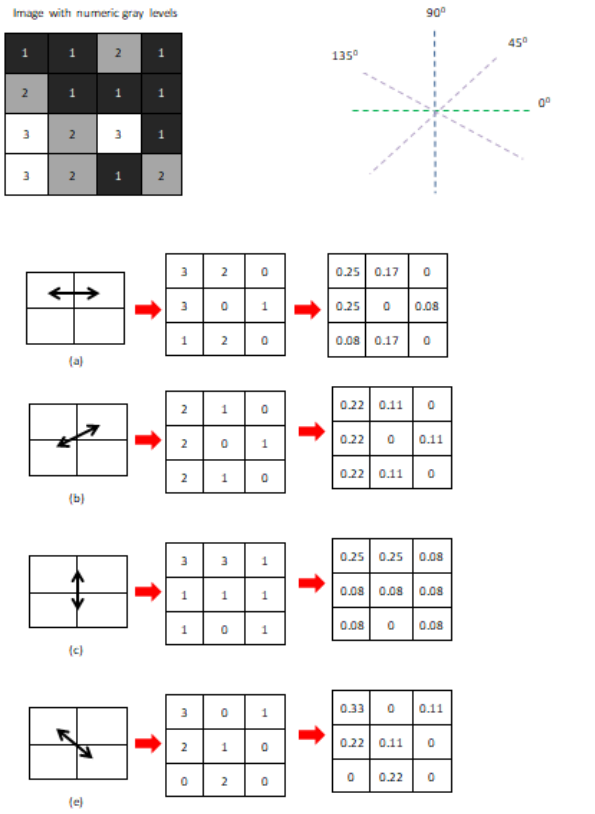



Figure 27: Computing the Haralick texture features from a 4 × 4 example image step by step

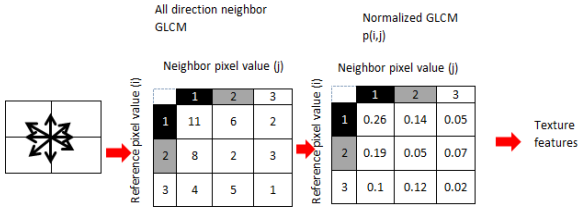



Figure 28: Computing the Haralick texture features from a 4 × 4 example image with all direction

Some of leaf images have very similar shape like Hathawariya (figure 29) and Iramusu (figure 29). Even though shapes are similar in some leaves, there are some differences in colors of leaf images. Therefore in addition to the shape features, we extracte color based features of leaf images as well.

We used mean($M$) and standard deviation($SD$) of intensity values of red, green and blue channels as color features.

$$M = \frac{\text{Total insensity value of } r^{th}\text{channel of the image pixels}}{\text{Total intensity value of the image}} \tag{11}$$

$$SD = \frac{\sqrt{\sum_{j=0}^{h}(r^{th}\text{channel intensity}_j - r^{th}\text{mean value})^2}}{\text{Total intensity value of the image}} \tag{12}$$

where $h$ is the number of pixels of the image and r is the channel type which can be red, green, or blue


Figure 29

## 3.4 Scagnostic features

Scagnostic features (Wilkinson, Anand, and Grossman 2005; Wilkinson and Wills 2008; Dang and Wilkinson 2014; Dang, Anand, and Wilkinson 2013) describe the characteristics of 2D scatterplot. Various types of measures are calculated based on the appearance of scatterplot. All the scagnostics features are calculated by using the R package called binostics. Scagnostic features has the range of [0,1]. There are 9 measures which are classified in to three categories as shown in figure 30. Based on binary images, scagnostic features are extracted.

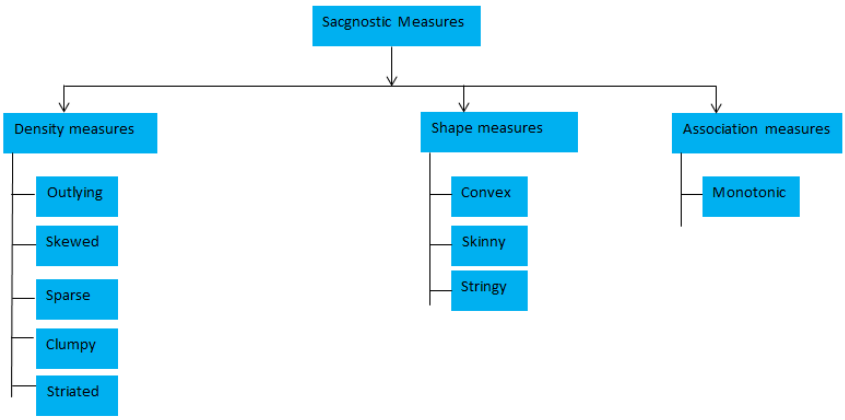



Figure 30: Hierarchy of Scagnostics

We measure the scagnostic features for cartesian and polar coordinate (see figure 31) separately.

As the first step, we have to extract the contour of the leaf image (see figure 32). Then find the x and y coordinate values of the cartesian and polar separately. The x and y coordinate value is used to calculate the scagnostic features.

The following definitions can be useful in understanding scagnostics features.

### 3.4.1 Geometric Graphs

- Graph

A graph Gr = (Ve, Ed) is defined as a set Ve (called vertices) together with a relation on Ve induced by a set Ed (called edges). A pair of vertices is defined as an edge $e(\nu, \omega)$, with e $\in$ Ed and $\nu, \omega \in$ Ve.

- Geometric Graph

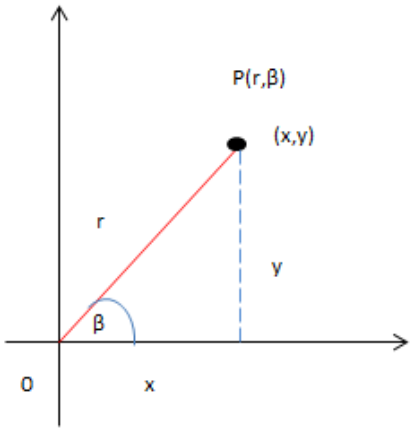



Figure 31: Polar coordinate

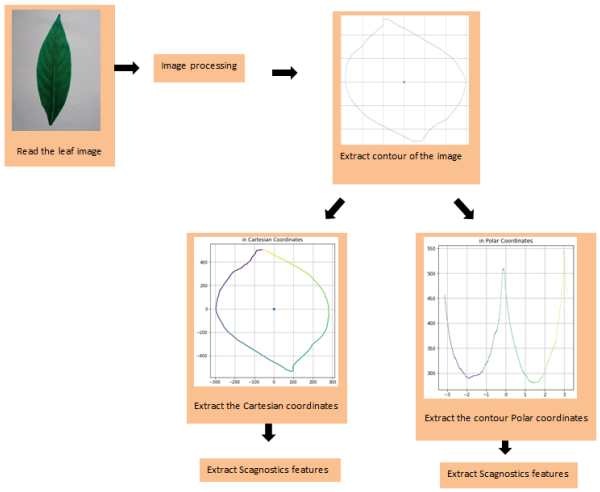



Figure 32: Preprocessing for Scagnostics

An embedding of a graph in a metric space S that maps vertices to points and edges to straight line segments connecting pairs of points is defined as a geometric graph G* = [f(Ve), g(Ed), S].

From several features of 2D Euclidean geometric graphs, Scagnostic measures are derived.

The Euclidean distance between vertices that connected to edge is defined as the length of an edge, length(e).

The sum of the lengths of edges in graph is known as the length of a graph, length(Gr).

A list of successively adjacent, distinct edges are known as a path. If first and last vertex are the same of the path, then the path is closed.

A region bounded by a closed path is known as a polygon (P). A polygon bounded by exactly one closed path that has no intersecting edges is known as a simple polygon.

The length of boundary of a simple polygon is known as the perimeter of a simple polygon. The area of interior of a simple polygon is known as the area of a simple polygon.

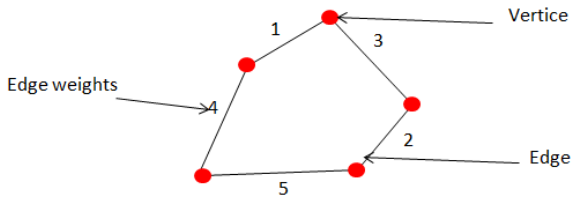



Figure 33: Graph with 5 vertices and 5 edges

### 3.4.2 Minimum Spanning Tree (MST)

- Tree

  A graph in which any two nodes are connected by exactly one path is known as a *tree*.

- Spanning Tree

  An undirected graph whose edges are structured as a tree is defines as a *Spanning Tree*.

  **Spanning Tree of Graph Gr is:** G'(V',E')

$$V' = Ve$$

$$E' \subset Ed$$

$$E' = |Ve| - 1$$

The graph can have more than one spanning tree.

Spanning tree should not be disconnected and not contain any cycle. By removing one edge from the Spanning tree will make it disconnected. By adding one edge to the Spanning tree will create a loop. A complete (Each vertices connected with each other) undirected graph can have $n^{n-2}$ number of spanning trees where n is the number of vertices. Every connected and undirected graph has at least one Spanning Tree. Disconnected graph doesn't have any spanning tree. From a complete graph by removing $max(edges - n + 1)$ edges we can construct a spanning tree.

- Minimum Spanning Tree

A spanning tree whose total length is least of all spanning trees on a given set of points is known as a Minimum Spanning Tree (MST).

If each edge has distinct weights then there will be only one and unique MST.

- Remark

The geometric MST computed from Euclidean distances between points in a 2D Euclidean geometric graph is the restriction.

### 3.4.3 Convex Hull

A collection of the boundaries of one or more simple polygons that have a subset of the points for their vertices and that collectively contain all the points, is defined as a hull of a set of points embedded in 2D Euclidean space.

If a hull contains all the straight line segments connecting any pair of points in its interior, is known as a convex hull. The convex hull bounds a single polygon. After deleting the points on the convex hull, a convex hull called peeled convex hull is computed.

### 3.4.4 Alpha Hull

Most of proximity graphs (neighborhood graph) represent the nonconvex shape of a set of points on the plane. A geometric graph whose edges are determined by an indicator function based on distances between a given set of points in a metric space, is known as a proximity graph. An open disk $D$ is used to define the indicator function.

If a point is on the boundary of $D$ then $D$ *touches* a point and if a point is in $D$ then $D$ *contains* a point. An open disk of radius r is defined as $D(r)$.

An alpha shape (Dang and Wilkinson 2014) is a collection of one or more simple polygons (Wilkinson and Wills 2008; Wilkinson, Anand, and Grossman 2005). An edge exists between any pair of points that can be touched by an open disk $D(\alpha)$ containing no points, is defined as an alpha shape graph.

A value of $\alpha$ to be the average value of the edge lengths in the MST (Wilkinson and Wills 2008; Wilkinson, Anand, and Grossman 2005). The large values like 90th percentile of the MST edge lengths are used, because to reduce noise. If the percentile exceeds a tenth, clamp the value at one-tenth the width of a frame, because it prevents in including sparse or striated point sets in a single alpha graph.

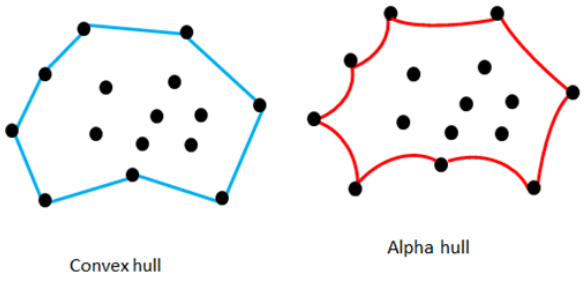



Figure 34: Convex hull and alpha hull

### 3.4.5 Preprocessing

To improve the performance of the algorithm and robustness of the measures, preprocessing techniques as binning and deleting outliers are used before computing geometric graphs.

- Binning

As the first step of binning, the data are normalized to the unit interval. Then use a 40 by 40 hexagonal grid to aggregate the points in each scatterplot. Reduce the bin size by half and rebin until no more than 250 non emty cells, if there are more than 250 non empty cells. By efficiency (too many bins slow down calculations of the geometric graphs) and sensitivity (too few bins obscure features in the scatterplots), the choice of bin size is constrained.

To improve the performance, hexagon binning is used. To manage the problem of having to many points that start to overlap, hexagon binning is used. The plots of hexagonal binning are density rather than points. To use hexagons instead of squares for binning a 2D surface as a plane, there are many reasons. Hexagons are more similar to circle than square.

To keep scagnostics orientation-independent this bias reduction is important. To attenuate the influence of binning, stabilizing transformation is used when computing scagnostics from binned data.

The weight function is defined as;

$$\text{weight} = 0.7 + \frac{0.3}{1+t^2} \tag{13}$$

where $t = \frac{n}{500}$. ($n$ is the number of vertex)

If $n > 2000$ then this function is fairly constant. By using hex binning the shape and the parameters of the function is determined. In computing Sparse, Skewed and Convex scagnostics this weight function is used to adjust for bias.

- Deleting Outliers

To improve robustness of the scagnostics, deleting outliers can be used. A vertex whose adjacent edges in the MST all have a weight (length) greater than $\omega$ is defined as an outlier in this context. By considering nonparametric criterion for the simplicity and Tukey's idea choose the following weight calculation.

$$\text{weight} = qu_{75} + 1.5(qu_{75} - qu_{25}) \tag{14}$$

where $qu_{75}$ is the 75th percentile of the MST edge lengths and $(qu_{75} - qu_{25})$ is the interquartile range of the edge lengths.

### 3.4.6 Degree of a Vertex

The degree of a vertex in an undirected graph is know as the number of edges associated with the vertex.

**Eg:- Vertices of degree 2** There are 2 edges associated with each vertex.

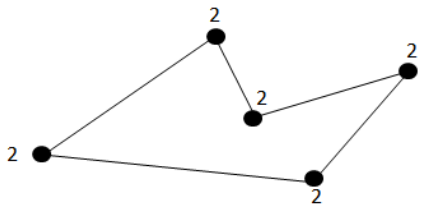



Figure 35: Vertices of degree 2

| Geometric Graphs | Notation |
|---|---|
| Convex Hull | CH |
| Alpha Hull | Al |
| Minimum Spanning Tree | T |

Table 4: Notations of Geometric Graphs

The definitions of scagnostic features are defined as follows.

### 3.4.7 Density Measures

Detect different distributions of scattered points in density measures.

- Outlying

$$F_{sc1} = \frac{\text{Total length of edges adjacent to outlying points}}{\text{Total edge length of T}} \tag{15}$$

The outlying measure calculate before deleting the outliers for the other measures.

- Skewed

$$qu_{skew} = \frac{qu_{90} - qu_{50}}{qu_{90} - qu_{10}} \tag{16}$$

$$F_{sc2} = 1 - \text{weight} * (1 - qu_{skew}) \tag{17}$$

where the weight function is equation 13.

The skewed measure is the first measure of relative density which is a relatively robust measure of skewness in the distribution of edge lengths. After adaptive binning skewed tends to decrease with $n$.

- Sparse

$$F_{sc3} = \text{weight} * qu_{90} \tag{18}$$

where the weight function is equation 13 and $qu_{90}$ is the 90th percentile of the distribution of edge lengths in the T.

The second relative density measure is sparse measure that measures whether points in a 2D scatterplot are confined to a lattice or a small number of locations on the plane.

If the number of points is extremely small or tuples are produced by the product of categorical variables, then sparse can be happen.

$$qu_{90} = \alpha statistic \tag{19}$$

The $\alpha$ statistic exceeds unity (e.g., when all points fall on either of the two diagonally opposing vertices of a square), clamp the value to 1 in the extremely rare event (Wilkinson and Wills 2008,@inproceedings44).

- Clumpy

$$F_{sc4} = max_j[1 - \frac{max_k[length(e_k)]}{length(e_j)}] \tag{20}$$

Clustering points are not indicated by an extreme distribution of T edge lengths. Therefore RUNT statistic (Wilkinson and Wills 2008; Wilkinson, Anand, and Grossman 2005) which is another measure based on the T, is introduced. The smaller of the number of leaves of each of the two subtrees joined at that node

isdefined as the runt size of a dendogram node. There is an association between runt size ($r_j$) each edge ($e_j$) in the T because there is an isomorphism between a single-linkage dendrogram and the T.

The smaller of the two subsets of edges that are still connected to each of the two vertices in $e_j$ after deleting edges in the MST with lengths less than length($e_j$), is known as the RUNT graph ($R_j$) (Wilkinson and Wills 2008; Wilkinson, Anand, and Grossman 2005).

The RUNT-based measure responds to clusters with small maximum intracluster distance relative to the length of their nearest-neighbor inter-cluster distance (Wilkinson and Wills 2008,@inproceedings44). In the formula j runs over all edges in T and k runs over all edges in RUNT graph.

- Striated

$$F_{sc5} = \frac{1}{|Ve|} \sum_{\nu \in Ve^{(2)}} I(\cos\theta_{e(\nu,a)e(\nu,b)} < -0.75) \tag{21}$$

where $Ve^{(2)} \subseteq Ve$ and $I()$ be an indicator function.

Striated define the coherence in a set of points as the presence of relatively smooth paths in the minimum spanning tree.

The measure is based on the number of adjacent edges whose cosine is less than minus 0.75.

### 3.4.8 Shape Measures

Both topological and geometric aspects of shape of a set of scattered points is considered. As an example, a set of scattered points on the plane appeared to be connected, convex and so forth, want to know under the shape measures. By definition scattered points are not like this. Therefore to make inferences additional machinery (based on geometric graphs) is needed. By measuring the aspects of the convex hull, the alpha hull, and the minimum spanning tree is determined.

- Convex

$$F_{sc6} = \text{weight} * \frac{\text{Area of alpha hull}}{\text{Area of convex hull}} \tag{22}$$

where the weight function is equation 13.

The ratio of the area of the alpha hall(Al) and the area of the convex hull(CH) is the base of measuring convexity.

$$Ratio = \frac{area(A)}{area(H)} \begin{cases} (=1); \text{ if the nonconvex hull and convex hull have identical areas} \\ \\ (\neq 1); \text{ otherwise} \end{cases}$$

‘

- Skinny

$$F_{sc7} = 1 - \frac{\sqrt{4 * \pi * \text{Area of alpha hull}}}{\text{Perimeter of alpha hull}} \tag{23}$$

Roughly, the skinny is measured by using the corrected and normalized ratio of perimeter to area of a polygon measures.

$$F_{sc7} = \begin{cases} \text{0; if circle} \\ \\ \text{Near 1; if skinny} \end{cases}$$

- Stringy

$$F_{sc8} = \frac{|Ve^{(2)}|}{|Ve| - |Ve^{(1)}|} \tag{24}$$

where $Ve$ is the number of vertices.

A skinny shape with no branches is known as a stringy shape. By counting the vertices of degree 2 in the minimum spanning tree and comparing them to the overall number of vertices minus the

number of single-degree vertices, skinny measure is calculated.

To adjust for negative skew in its conditional distribution of $n$, cube the stringy measure.

#### 3.4.9 Association Measure

Symmetric and relatively robust measure of association are interested.

- Monotonic

$$F_{sc9} = r^2_{Spearman} \tag{25}$$

To assess the monotonicity in a scatter plot, the squared Spearman correlation coefficient is used. This is the only coefficient not based on a subset of the Delaunay graph (Wilkinson and Wills 2008).

In calculating monotonicity, squared the coefficient because to consider the large values and to remove the distinction between positive and negative coefficients (Because assume that the investigators are more interested in strong relationships rather than negative or positive).

We introduced number of minimum and maximum points, correlation of cartesian contour as new features under scagnostics.

### 3.5 Number of Minimum ($F_8$) and Maximum Points ($F_9$)

Number of minimum, and maximum points are new measures which are obtained from the polar coordinate of leaf contour. Number of global maximum points are defined as number of maximum points. Number of global minimum points are defined as number of minimum points.

$$F_8 = \text{Number of global minimum points} \tag{26}$$

$$F_9 = \text{Number of global maximum points} \tag{27}$$

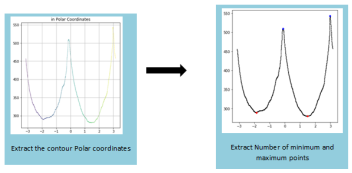



Figure 36: Minimum and Maximum Points

### 3.6 Correlation of Cartesian Contour ($F_{10}$)

Correlation is another new feature which is obtained from the cartesian contour.

$$F_{10} = \frac{\sum_{i=0}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=0}^{m}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{28}$$

where $(x_i, y_i)$ is the coordinate of cartesian contour and $m$ is the number of points in the cartesian contour

## 4 Application of features to classify images

There are two color image leaf datasets (Flavia ,Swedish) are used. After passing through the required image processing steps, binary image is extracted which has a white foreground and black background.

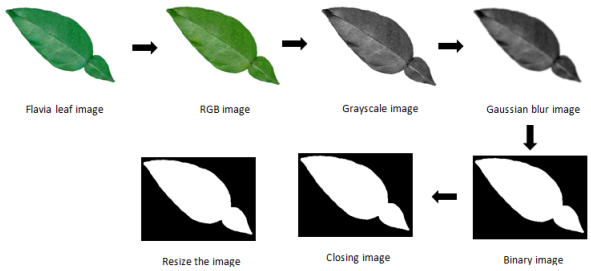



Figure 37: Image processing steps of Flavia dataset

The leaf images are taken as the closest ones. Therefore to find the best contour among several contours, can use the contour which contains the center of leaf image. Identify the best contour is really important when extracting shape features.

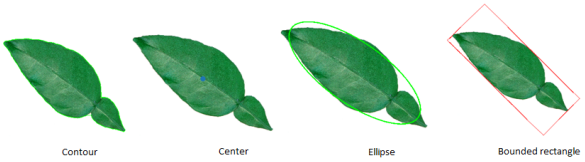



Figure 38: Example feature extraction of Flavia dataset

### 4.1 Data sets

We use two publicly available datasets for demonstrate the applications of features.

### 4.1.1 Flavia Leaf Image Dataset

The Flavia dataset contains 1907 leaf images. There are 32 different species and each have 50-77 images. Scanners and digital cameras are used to acquire the leaf images on plain background. The isolated leaf images contain blades only, without petiole. These leaf images are collected from the most common plants in Yangtze, Delta, China (Waldchen and Mader 2018). Those leaves were sampled on the campus of the Nanjing University and the Sun Yat-Sen arboretum, Nanking, China (Waldchen and Mader 2018). (https://sourceforge.net/projects/flavia/files/Leaf%2520Image%2520Dataset/)

Figure 39: Sample of Flavia dataset

### 4.1.2 Swedish Leaf Image Dataset

The Swedish dataset contains 1125 images. The images of isolated leaf scans on a plain background of 15 Swedish tree species, with 75 leaves per species. This dataset has been captured as part of a joined leaf classification project between the Linkoping University and the Swedish Museum of Natural History (Waldchen and Mader 2018). (https://www.cvl.isy.liu.se/en/research/datasets/swedish-leaf/)

Figure 40: Sample of Swedish leaf dataset

## 4.2 Visualization of Leaf Images in the Feature Space

We used LDA, and PCA to visualize leaf images in feature space. LDA is a supervised dimensionality reduction technique, and PCA is unsupervised dimensionality reduction technique. In this section, we visualize, and compare the results obtained using LDA, and PCA on flavia and swedish datasets. To visualize LDA projection shape label is taken as the response variable. There are mainly 5 shape categories as diamond, simple round, round, needle, and heart shape. More details about the shape categories are discussed in our next paper.

### 4.2.1 Swedish Dataset

#### PCA Projection

The results of the PCA1 and PCA2, and PCA1 and PCA3 projection on swedish dataset shows clear separation among the shapes.

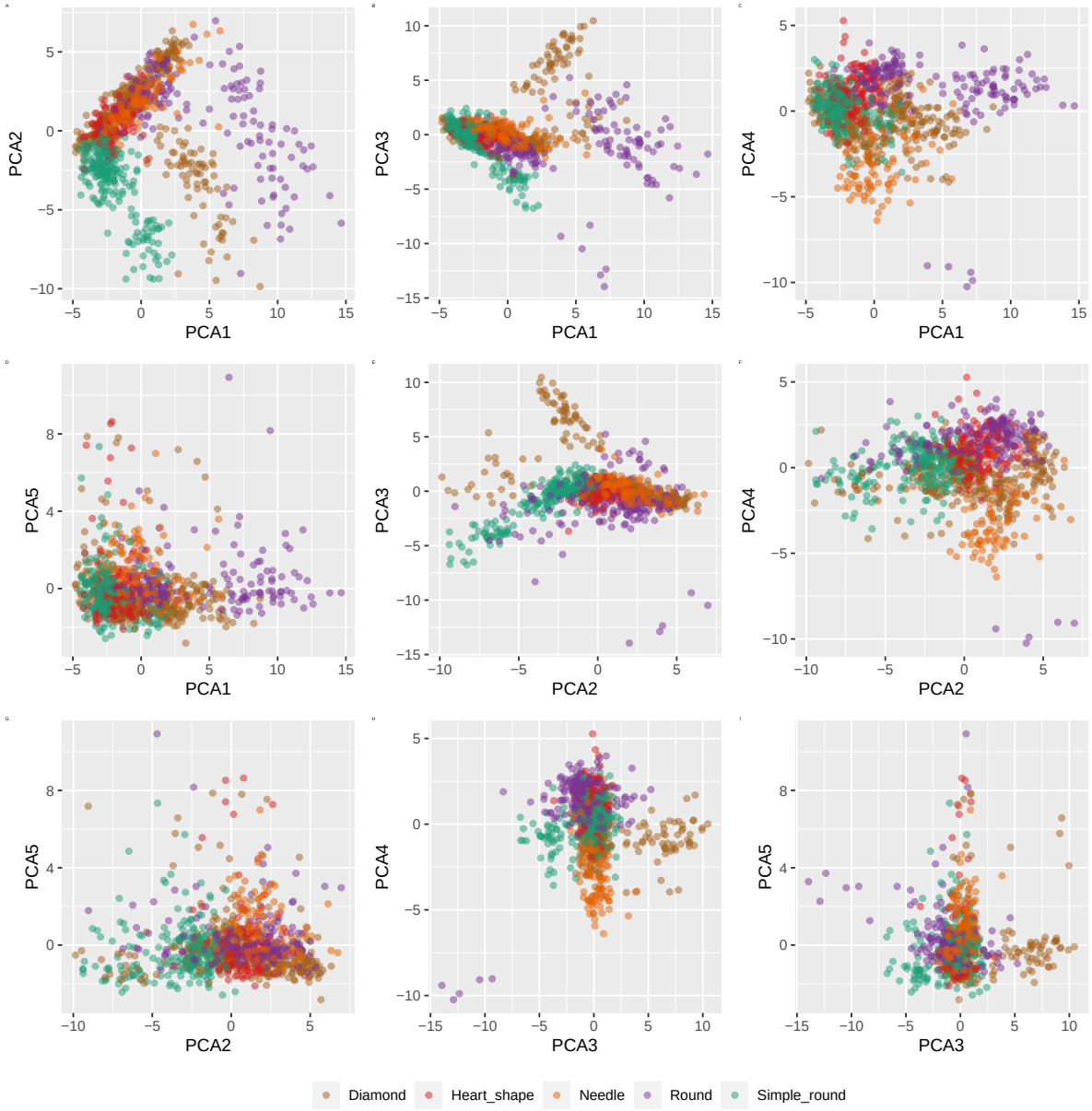



Figure 41: Distribution of Swedish leaf images on the projection space created based on pincipal component analysis. All projected points are coloured according to their shape labels.

#### LDA Projection

When consider the LDA results on swedish dataset, LDA1, LDA2, and LDA3 shows the separation of shapes of leaf images.

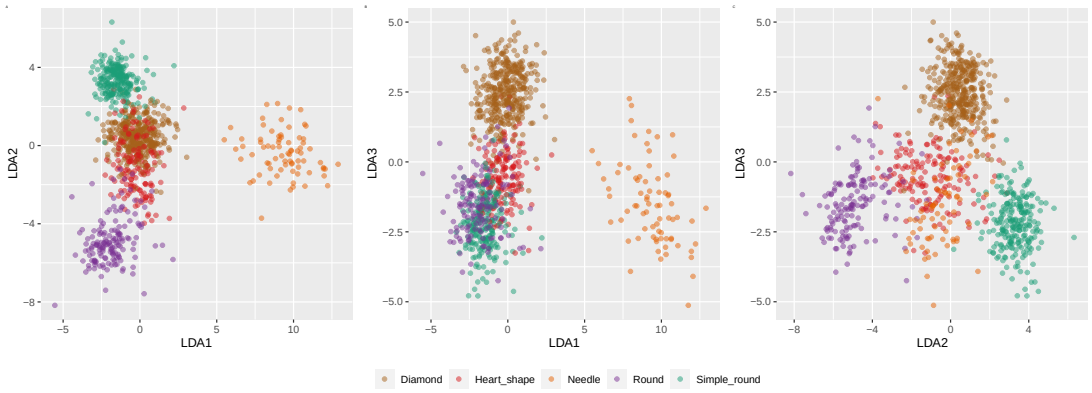



Figure 42: Distribution of Swedish leaf images on the projection space created based on linear discriminant analysis. All projected points are coloured according to their shape labels.

### 4.2.2 Flavia Dataset

**PCA projection**

**LDA Projection**

The projected results of LDA1 and LDA2 on flavia dataset shows clear classification of leaf shapes than PCAs.

# 5 Discussion and Conclusions

There are mainly four categories of features that are used to classify leaf images. Many research were based on shape, color, and texture features. In this research paper, we introduced new feature category called scagnostics. Other than that correlation of cartesian coordinate, number of convex points, number of minimum and maximum points are introduced as new shape features. To visualize the leaf images on feature space LDA, and PCA are used. The results are obtained based on flavia and swedish datasets. LDA is performed better than PCA when classifying leaf shapes. More details about the classification of leaf images are described in our next paper.

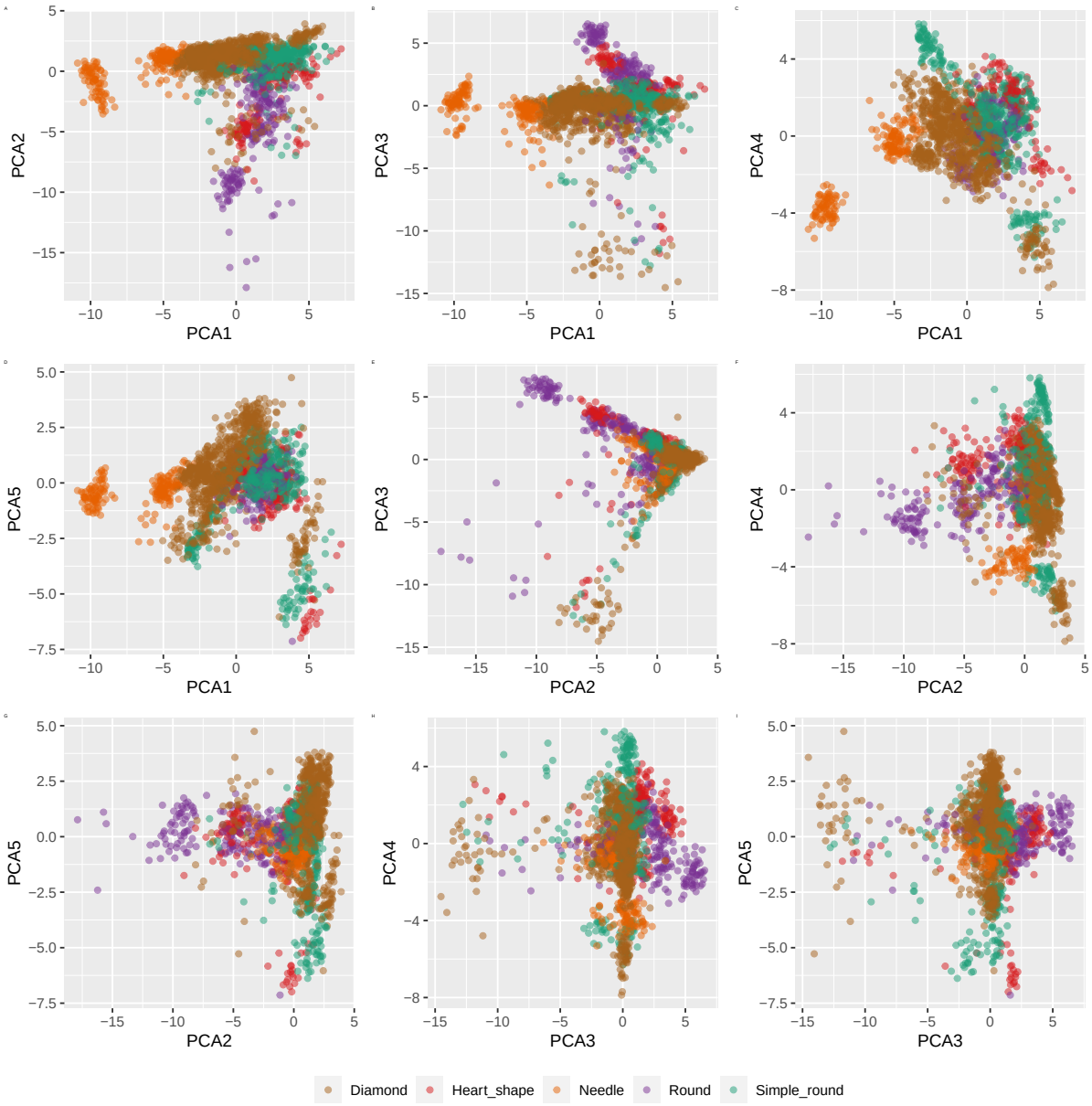



Figure 43: Distribution of Flavia leaf images on the projection space created based on pincipal component analysis. All projected points are coloured according to their shape labels.

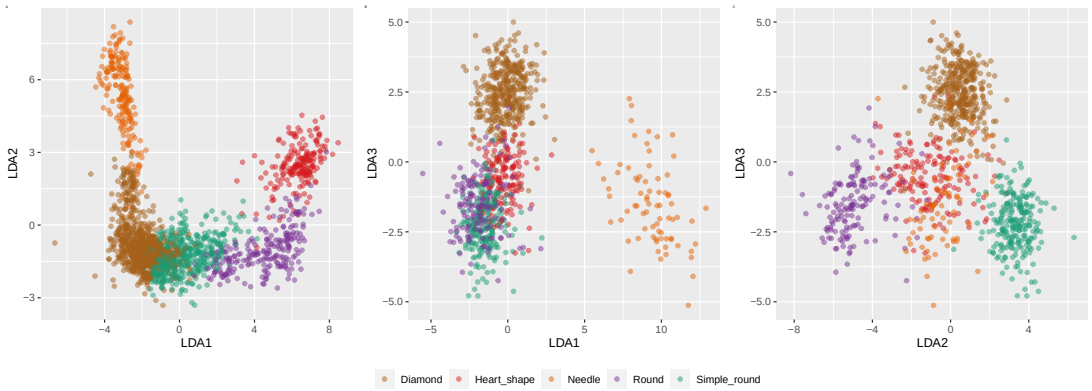



Figure 44: Distribution of Flavia leaf images on the projection space created based on linear discriminant analysis. All projected points are coloured according to their shape labels.

# Reference

Anantrasirichai, Nantheera, Sion L. Hannuna, and Nishan Canagarajah. 2017. "Automatic Leaf Extraction from Outdoor Images." *CORR* abs/1709.06437. http://arxiv.org/abs/1709.06437.

Azlah, Muhammad, Lee Suan Chua, Fakhrul Rahmad, Farah Abdullah, and Sharifah Alwi. 2019. "Review on Techniques for Plant Leaf Classification and Recognition." *COMPUTERS* 8 (October): 77. https://doi.org/10.3390/computers8040077.

Bangare, Sunil L, Amruta Dubal, Pallavi S Bangare, and ST Patil. 2015. "Reviewing Otsu's Method for Image Thresholding." *International Journal of Applied Engineering Research* 10 (9): 21777–83.

Boland, MV. 2000. "Quantitative Description and Automated Classification of Cellular Protein Localization Patterns in Fluorescence Microscope Images of Mammalian Cells."

Caglayan, Ali, Oguzhan Guclu, and Ahmet Can. 2013. "A Plant Recognition Approach Using Shape and Color Features in Leaf Images." In, 161–70. https://doi.org/10.1007/978-3-642-41184-7_17.

Dang, T. N., A. Anand, and L. Wilkinson. 2013. "TimeSeer: Scagnostics for High-Dimensional Time Series." *IEEE Transactions on Visualization and Computer Graphics* 19 (3): 470–83. https://doi.org/10.1109/TVCG.2012.128.

Dang, Tommy, and Leland Wilkinson. 2014. "ScagExplorer: Exploring Scatterplots by Their Scagnostics." In *IEEE Pacific Visualization Symposium*, 73–80. https://doi.org/10.1109/PacificVis.2014.42.

Gonzalez, Rafael C., and Richard E. Woods. 2006. *Digital Image Processing (3rd Edition)*. USA: Prentice-Hall, Inc.

Goyal, N., Kapil, and N. Kumar. 2018. "Plant Species Identification Using Leaf Image Retrieval: A Study." In *2018 International Conference on Computing, Power and Communication Technologies (Gucon)*, 405–11.

Haralick, Robert, K. Shanmugam, and Ih Dinstein. 1973. "Textural Features for Image Classification." *IEEE Trans Syst Man Cybern* SMC-3 (January): 610–21.

Herdiyeni, Yeni, and Ni Wahyuni. 2012. "Mobile Application for Indonesian Medicinal Plants Identification Using Fuzzy Local Binary Pattern and Fuzzy Color Histogram." In *2012 INTERNATIONAL CONFERENCE ON ADVANCED COMPUTER SCIENCE AND INFORMATION SYSTEMS, ICACSIS 2012 - PROCEEDINGS*.

Jeon, Wang-Su, and Sang-Yong Rhee. 2017. "Plant Leaf Recognition Using a Convolution Neural Network." *THE INTERNATIONAL JOURNAL OF FUZZY LOGIC AND INTELLIGENT SYSTEMS* 17 (March): 26–34. https://doi.org/10.5391/IJFIS.2017.17.1.26.

Kodituwakku, Saluka, and S.Selvarajah. 2010. "Comparison of Color Features for Image Retrieval." *Indian Journal of Computer Science and Engineering* 1 (October).

Söderkvist, Oskar. 2001. "Computer Vision Classification of Leaves from Swedish Trees."

Sun, Yu, Yuan Liu, Guan Wang, and Haiyan Zhang. 2017. "Deep Learning for Plant Identification in Natural Environment." *Computational Intelligence and Neuroscience* 2017. Hindawi.

Waldchen, Jana, and Patrick Mader. 2018. "Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review." *ARCHIVES OF COMPUTATIONAL METHODS IN ENGINEERING* 25 (April): 507–43. https://doi.org/10.1007/s11831-016-9206-z.

Wilkinson, Leland, A. Anand, and Robert Grossman. 2005. "Graph-Theoretic Scagnostics." In *INFOVIS*, 5:157–64. https://doi.org/10.1109/INFVIS.2005.1532142.

Wilkinson, Leland, and Graham Wills. 2008. "Scagnostics Distribution." *Journal of Computational and Graphical Statistics - J COMPUT GRAPH STAT* 17 (June): 473–91. https://doi.org/10.1198/106186008X320465.

Wu, S. G., F. S. Bao, E. Y. Xu, Y. Wang, Y. Chang, and Q. Xiang. 2007. "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network." In *2007 Ieee International Symposium on Signal Processing and Information Technology*, 11–16.

Wu, Stephen Gang, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. 2007. "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network." In *2007 Ieee International Symposium on Signal Processing and Information Technology*, 11–16. IEEE.