

## EXPERIMENTS OF ZERNIKE MOMENTS FOR LEAF IDENTIFICATION

<sup>1</sup>ABDUL KADIR, <sup>2</sup>LUKITO EDI NUGROHO, <sup>3</sup>ADHI SUSANTO, <sup>2</sup>P. INSAP SANTOSA

<sup>1</sup>Ph.D. Student, Department of Electrical Engineering, Gadjah Mada University, Indonesia

<sup>2</sup>Lecturer, Department of Electrical Engineering, Gadjah Mada University, Indonesia

<sup>3</sup>Professor, Department of Electrical Engineering, Gadjah Mada University, Indonesia

<sup>4</sup>Lecturer, Department of Electrical Engineering, Gadjah Mada University, Indonesia

E-mail: <sup>1</sup>[akadir@mti.ugm.ac.id](mailto:akadir@mti.ugm.ac.id), <sup>2</sup>[lukito@mti.ugm.ac.id](mailto:lukito@mti.ugm.ac.id), <sup>3</sup>[susanto@te.ugm.ac.id](mailto:susanto@te.ugm.ac.id), <sup>4</sup>[insap@mti.ugm.ac.id](mailto:insap@mti.ugm.ac.id)

### ABSTRACT

So far, plant identification has challenges for several researchers. Various methods and features have been proposed. However, there are still many approaches could be investigated to develop robust plant identification systems. This paper reports several experiments in using Zernike moments to build foliage plant identification systems. In this case, Zernike moments were combined with other features: geometric features, color moments and gray-level co-occurrence matrix (GLCM). To implement the identifications systems, two approaches has been investigated. First approach used a distance measure and the second used Probabilistic Neural Networks (PNN). The results show that Zernike Moments have a prospect as features in leaf identification systems when they are combined with other features.

**Keywords:** *Zernike Moments, Leaf identification system, GLCM, PNN, City block.*

### 1. INTRODUCTION

Identifying plants by using leaf image has been explored by several researches [1]-[8]. Various approaches has been proposed to build plant identification systems. Some of researches focused on green leaves and ignored color information on the leaf. For example, Wu et al. [1] used PNN to classify 32 kinds of plants. All the plants were used in their research have green leaves. Zulkifli [2] did a research using General Regression Neural Networks (GRNN) and invariant moment to classify 10 kinds of plants. The research also did not include color features to the classifier. However, the color information has been incorporated in the plant identification by Man et al. [3]. They reported that their identification system could recognize 24 kinds of plants with accuracy up to 92.2%.

In order to obtain high performance of the leaf identification systems, combination of methods and features has been introduced. Chaki and Parakh [4] proposed a method called binary superposition that based on shape features only. The approach compared the binary versions of the leaf images through superposition and using the sum of non-zero pixel values of the resultant as the feature

vector. The average accuracy was 99% for three plants: *Arbutus unedo*, *Betula pendula* and *Pittosporum\_tobira*. Wu et al. [5] used several geometric features and invariant moments to capture shape of the leaf and other features to capture edge of the leaf and vein of the leaf for recognizing 6 kinds of plants. The accuracy was between 91.1% and 98.6%. Ehsanirad and Kumar [6] developed a plant recognition system by using gray level co-occurrence matrix (GLCM) and Principal Component Analysis (PCA) to classify 13 kinds of plants. The accuracy was 98.46%. Jiming [7] proposed a method called neighborhood rough set to classify 30 kinds of plants and gave 95.83% of accuracy. Singh et al. [8] did a research that used dataset Flavia came from [1]. They reported that the accuracy were 46,30% by using Fourier moments, 91,25% by using PNN-PCNN, and 81.56% by using Support Vector Machine (SVM) in classifying 32 kinds of plant. Lee and Chen [9] classified 60 kinds of plants by using 1-NN with 82.33% of accuracy.

In this research, 60 kinds of leaves came from foliage plants had been tested. Beside shape, color and texture were considered to obtained high accuracy. It was dedicated to handle two or more plants that have similar/same shape but the color

patterns on the leaves were different. Such condition occurs for example in *Epripremnum pinnatum* "Aureum" and *Epripremnum pinnatum* "Marble Queen". All features used in the identification systems, either using a distance measure or Probabilistic Neural Networks (PNN).

## 2. BASIC OF THE SYSTEM

### 2.1 Geometric Features

Geometric features are commonly used in identification systems. Several geometric features that were used in this research are described here.

**Aspect ratio:** Aspect ratio or sometimes called as eccentricity is defined as ratio between length of the leaf minor axis (w) and the length of the leaf major axis (Figure 1). It is notated as

$$\text{aspect ratio} = \frac{w}{l}$$

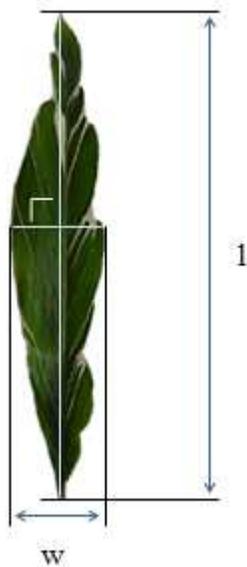


Figure 1. Components for calculating aspect ratio.

**Circularity:** Circularity is ratio between area of the leaf (a) and square of perimeter (p) of the leaf. It can be notated as

$$\text{circularity} = \frac{a}{p^2}$$

Figure 2 shows the area of a leaf and its perimeter.

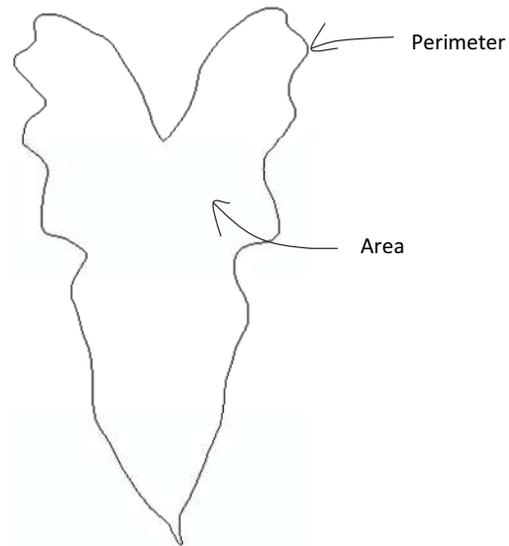


Figure 2. Area and perimeter of a leaf.

**Irregularity:** Irregularity or dispersion is defined as ratio between the radius of the maximum circle enclosing the region and the minimum circle that can be contained in the region [10]. It is notated as

$$\text{irregularity} = \frac{\max(\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2})}{\min(\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2})}$$

where  $(\bar{x}, \bar{y})$  is the centroid of the leaf and  $(x_i, y_i)$  is the coordinate of a pixel in the leaf contour.

**Vein features:** Vein features are features derived from vein of the leaf. There were two kinds of vein features, defined as follows:

$$V_1 = \frac{A_1}{A}$$

$$V_2 = \frac{A_2}{A}$$

$A_1$  and  $A_2$  are pixel number that constructs the vein and  $A$  is area of the leaf. The vein of the leaf was constructed by using morphological operation called opening [1]. The example of the results is shown in Figure 3.

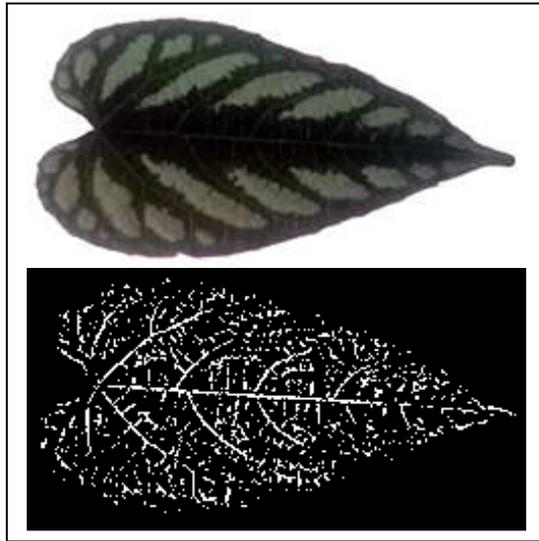


Figure 3. Leaf and the structure like vein.

The algorithm used in that operation is described as follow.

1. Convert RGB image of the leaf to gray scale image and binary image:

$$\text{Grayscale\_leaf} \leftarrow \text{grayscale of the leaf}$$

$$\text{Binary\_leaf} \leftarrow \text{binary of the leaf}$$

2. Create two structure elements with disk shape and the radius 1 and 2 respectively:

$$\text{SE}_1 \leftarrow \text{disk shape structure element, } r=1$$

$$\text{SE}_2 \leftarrow \text{disk shape structure element, } r=2$$

3. Do opening operation to both structure elements:

$$B_1 \leftarrow \text{opening}(\text{SE}_1)$$

$$B_2 \leftarrow \text{opening}(\text{SE}_2)$$

4. Subtracted the grayscale images of the leaf by the result of the opening operation:

$$D_1 \leftarrow \text{subtract}(\text{Grayscale\_leaf}, B_1)$$

$$D_2 \leftarrow \text{subtract}(\text{Binary\_leaf}, B_2)$$

5. Get the binary images of  $D_1$  and  $D_2$  by using Otsu method:

$$BW_1 \leftarrow \text{Segmented\_by\_Otsu}(D_1)$$

$$BW_2 \leftarrow \text{Segmented\_by\_Otsu}(D_2)$$

6. Calculate the total of pixels of  $BW_1$  and  $BW_2$ :

$$V_1 \leftarrow \text{pixel number of } BW_1$$

$$V_2 \leftarrow \text{pixel number of } BW_2$$

7. Calculate area of the leaf:

$$A \leftarrow \text{area of}(\text{Binary leaf})$$

8. Calculate the ratio of:

$$A_1 \leftarrow V_1 / A$$

$$A_2 \leftarrow V_2 / A$$

**Solidity:** solidity is defined as ratio between the area of the leaf and the area of its convex hull [11]. So,

$$\text{solidity} = \frac{\text{area of leaf}}{\text{area of convex}}$$

**Convexity:** Convexity is defined as ratio between the convex hull perimeter of the leaf and the perimeter of the leaf [11]. Mathematically, it is notated as

$$\text{convexity} = \frac{\text{convex perimeter}}{\text{perimeter}}$$

Figure 4 shows an example of the convex hull (the red color).

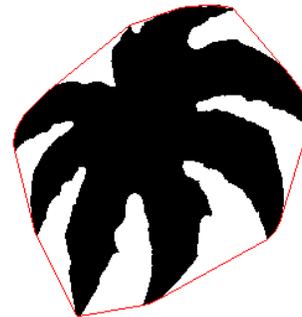


Figure 4. Convex hull.

Convex hull can be obtained by using algorithm 'Graham Scan' [12]. The basic principle to obtain a convex hull is described below.

1. Find a point  $p_0$  from a set P that contains set of pixels. This point is known as an anchor point. To obtain this point, find a pixel in P that has minimum value in its ordinate (Y). If there several pixel fulfill the condition, find a pixel that has minimum value of absis (X).

2. Calculate the angle of each point in the set P, except the p<sub>0</sub>, to point p<sub>0</sub>. Then, the points are sorted radially counter the clockwise (Figure 5).

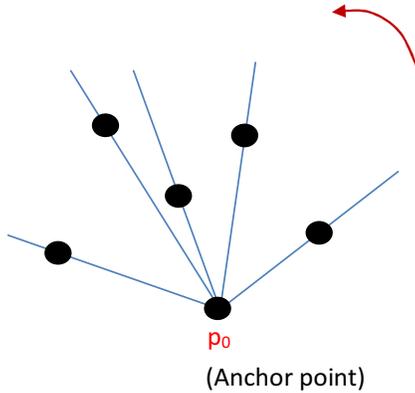


Figure 5. Sort the points radially counter the clockwise.

3. If there are more than one point that have same angle, delete all those points except a point that has the longest distance from p<sub>0</sub>.
4. Prepare a stack, called H. Point p<sub>0</sub> and the first point in the result of sorting process is pushed to the stack. Then, the remaining points (p<sub>i</sub>) is processed as follow.
  - (a) If p<sub>i</sub> build a left turn toward two top points in the stack H, insert the p<sub>i</sub> to stack H and process the next point.
  - (b) Otherwise, pop a point from the stack H.

The algorithm of 'Graham Scan' is described below.

Input: P = n pixels

Output: Set of convex hull

GrahamScan(P):

1. P<sub>0</sub> ← anchor point
2. P ← P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, ..., P<sub>n-1</sub> where P<sub>1</sub>..P<sub>n-1</sub> had been sorted radially.
3. H ← empty stack
4. Push(H, p<sub>0</sub>)
5. Push(H, p<sub>1</sub>)
6. i ← 2
7. WHILE i ≤ n
  - p<sub>a</sub> ← top of H
  - p<sub>b</sub> ← top of H

```

IF pi on the right of line(Pa, Pb)
    Pop(H)
ELSE
    Push(H, Pi)
    i ← i + 1
END
END
END
    
```

8. RETURN H

The expression p<sub>i</sub> on the right of line(P<sub>a</sub>, P<sub>b</sub>) can be calculated as follow:

```

On_the_right_of_line(pa, pb, p):
    ((pb.x - pa.x) * (p.y - pa.y) -
    (p.x - pa.x) *
    (pb.y - pa.y)) > 0;
    
```

**Color Moments:** Color moments can be extracted from R,G and B component on the leaf by using statistical calculations such mean (μ), standard deviation (σ), skewness (θ) and kurtosis (γ) [13]. The 4 features per R, G and B components were calculated using the following formulas:

$$\mu = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N P_{ij}$$

$$\sigma = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^2}$$

$$\theta = \frac{\sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^3}{MN\sigma^3}$$

$$\gamma = \frac{\sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^4}{MN\sigma^4} - 3$$

**GLCM:** Gray level co-occurrence matrix (GLCM) is usually used to capture texture features in several researches such as in [14] and [15]. GLCM is computed in grayscale image as below [15]

$$GLCM(i, j) = \# \{ | (x, y), (x + d, y + d) \in S | f(x, y) = i, f(x + d, y + d) = j \} / \# S$$

where:

- $x, y = 1, 2, \dots, N$  are coordinates of the pixels
- $i, j = 0..L-1$  are the gray levels
- $d$  is the distance between two pixels
- $S$  is set of pixel pairs
- $\#S$  is number of elements in  $S$
- $GLCM(i, j)$  is the probability density that the pixel with gray level  $i$  and next pixel with distance is  $d$  and the gray level  $j$ .

In this research, the distance between two pixels was one and the directions were used: 0, 45, 90, and 135 degrees. Therefore, there were 4 GLCMs.

Based on each GLCM, the following features were computed: angular second moment (ASM), contrast, inverse different moment (IDM), entropy, and correlation. Then, five features were obtained by averaging them. This action was accomplished to obtain rotation independence. So, there were 5 features to capture the texture of the leaf.

The ASM (or energy) measures textural uniformity [16]. It is computed as

$$ASM = \sum_{i=1}^L \sum_{j=1}^L (GLCM(i, j))^2$$

The contrast determines the coarse texture or variance of the grey level. It is computed as

$$Contrast = \sum_{i=1}^L \sum_{j=1}^L (i - j)^2 (GLCM(i, j))$$

The IDM measures the local homogeneity pair of pixels. It is computed as

$$IDM = \sum_{i=1}^M \sum_{j=1}^N \frac{(GLCM(i, j))^2}{1 + (i - j)^2}$$

The entropy measures the degree of disorder of image. It is computed as

$$Entropy = - \sum_{i=1}^L \sum_{j=1}^L GLCM(i, j) \log(GLCM(i, j))$$

The correlation texture measures the linear dependency of gray levels on those of neighboring pixels. It is computed as

$$Correlation = \sum_{i=1}^L \sum_{j=1}^L \frac{(ij)(GLCM(i, j) - \mu_i' \mu_j')}{\sigma_i' \sigma_j'}$$

where

$$\mu_i' = \sum_{i=1}^L \sum_{j=1}^L i * GLCM(i, j)$$

$$\mu_j' = \sum_{i=1}^L \sum_{j=1}^L j * GLCM(i, j)$$

$$\sigma_i'^2 = \sum_{i=1}^L \sum_{j=1}^L GLCM(i, j) (i - \mu_i')^2$$

$$\sigma_j'^2 = \sum_{i=1}^L \sum_{j=1}^L GLCM(i, j) (j - \mu_j')^2$$

In five previous equations,  $L$  is number of rows/columns in GLCM.

**Zernike Moments:** Zernike moments are based on Zernike polynomials that are orthogonal to the circle  $x^2 + y^2 \leq 1$  [17] (Figure 6). It can be notated as below:

$$V_{pq}(x, y) = U_{pq}(r \cos \theta, r \sin \theta) \\ = R_{pq}(r) \exp(jq\theta)$$

where  $r$  is the radius of  $(x, y)$  to the centroid,  $\theta$  is the angle between  $r$  and axis  $x$ , and  $R_{pq}(r)$  is the polynomial defined as

$$R_{pq}(r) = \sum_{s=0}^{(p-|q|)/2} (-1)^s \frac{(p-s)!}{s! \binom{p+|q|}{2} \binom{p-|q|}{2} s!} r^{p-2s}$$

In this case,  $n=0, 1, 2, \dots$ ;  $0 \leq |q| \leq n$ ,  $j = \sqrt{-1}$ , and  $p-|q|$  is even.

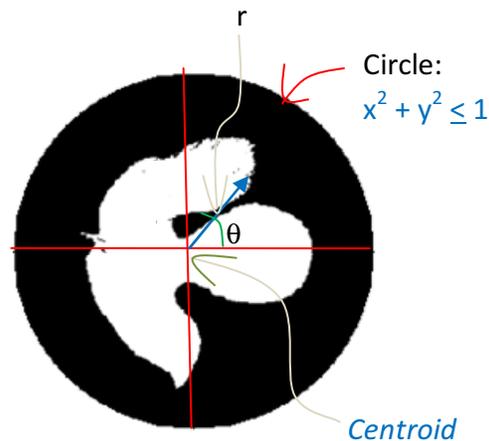


Figure 6. Image in the circle fulfills  $x^2 + y^2 \leq 1$ .

Zernike moment order  $p$  with  $q$  repetition of continued function  $f(y, x)$  is defined as below:

$$Z_{pq} = \frac{p+1}{\pi} \iint_{x,y} f(x,y) \cdot V^*(x,y) dy dx; x^2 + y^2 \leq 1$$

In this case,  $V^*$  is the complex conjugate, whereas  $V_{pq}(y,x)$  is called as a Zernike basis function order  $p$  with  $q$  repetitions. The Zernike basis function is

$$V_{pq}(x,y) = V_{pq}(\rho, \theta) = R_{pq}(\rho) \cdot \exp(jq\theta)$$

where  $p$  is zero or positive integer,  $p-|q|$  is odd and  $|q| \leq p$ . If  $f(x,y)$  is a digital image, Zernike moment can be approximate as [17]

$$Z_{pq} = \frac{p+1}{\pi} \sum_x \sum_y f(x,y) \cdot V^*_{pq}(x,y)$$

The polynomials that can be used to calculate Zernike moments from order 2 to 12 are shown in Table 1 [18][19].

Table 1. Zernike polynomials.

Order	Polynomials
2	$R_{2,0}(r) = 2r^2 - 1$ $R_{2,2}(r) = r^2$
3	$R_{3,1}(r) = 3r^2 - 2r$ $R_{3,3}(r) = r^3$
4	$R_{4,0}(r) = 6r^4 - 6r^2 + 1$ $R_{4,2}(r) = 4r^4 - 3r^2$ $R_{4,4}(r) = r^4$
5	$R_{5,1}(r) = 10r^5 - 12r^2 + 3r$ $R_{5,3}(r) = 5r^5 - 4r^3$ $R_{5,5}(r) = r^5$
6	$R_{6,0}(r) = 20r^6 - 30r^4 + 12r^2 - 1$ $R_{6,2}(r) = 15r^6 - 20r^4 + 6r^2$ $R_{6,4}(r) = 6r^6 - 5r^4$ $R_{6,6}(r) = r^6$

Order	Polynomials
7	$R_{7,1}(r) = 35r^6 - 60r^5 + 30r^3 - 4r$ $R_{7,3}(r) = 21r^7 - 30r^5 + 10r^3$ $R_{7,5}(r) = 7r^7 - 6r^5$ $R_{7,7}(r) = 7r^7$
8	$R_{8,0}(r) = 70r^8 - 140r^6 + 90r^4 - 20r^2 + 1$ $R_{8,2}(r) = 56r^8 - 105r^6 + 60r^4 - 10r^2$ $R_{8,4}(r) = 28r^8 - 42r^6 + 15r^4$ $R_{8,6}(r) = 8r^8 - 7r^6$ $R_{8,8}(r) = r^8$
9	$R_{9,1}(r) = 126r^9 - 280r^7 + 210r^5 + 60r^3 + 5r$ $R_{9,3}(r) = 84r^9 - 168r^7 + 105r^5 - 20r^3$ $R_{9,5}(r) = 36r^9 - 56r^7 + 21r^5$ $R_{9,7}(r) = 9r^9 - 8r^7$ $R_{9,9}(r) = r^9$
10	$R_{10,0}(r) = 252r^{10} - 630r^8 + 560r^6 - 210r^4 + 30r^2 + 5r$ $R_{10,2}(r) = 210r^{10} - 504r^8 + 420r^6 - 140r^4 + 15r^2$ $R_{10,4}(r) = 129r^{10} - 252r^8 + 168r^6 - 35r^4$ $R_{10,6}(r) = 45r^{10} - 72r^8 + 28r^6$ $R_{10,8}(r) = 10r^{10} - 9r^8$ $R_{10,10}(r) = r^{10}$

Order	Polynomials
11	$R_{11,1}(r) = 462r^{11} - 1260r^9 + 1260r^7 - 560r^5 + 105r^3 - 6r$ $R_{11,3}(r) = 330r^{11} - 840r^9 + 757r^7 - 280r^5 + 35r^3$ $R_{11,5}(r) = 165r^{11} - 360r^9 + 252r^7 - 56r^5$ $R_{11,7}(r) = 55r^{11} - 90r^9 + 36r^7$ $R_{11,9}(r) = 11r^{11} - 10r^9$ $R_{11,11}(r) = r^{11}$
12	$R_{12,0}(r) = 924r^{12} - 2772r^{10} + 3150r^8 - 1680r^6 + 420r^4 - 42r^2 + 1$ $R_{12,2}(r) = 792r^{12} - 2310r^{10} + 2520r^8 - 1260r^6 - 504r^4 + 70r^2$ $R_{12,4}(r) = 495r^{12} - 1320r^{10} + 1260r^8 - 504r^6 + 70r^4$ $R_{12,6}(r) = 220r^{12} - 495r^{10} + 360r^8 - 84r^6$ $R_{12,8}(r) = 66r^{12} - 110r^{10} + 45r^8$ $R_{12,10}(r) = 12r^{12} - 11r^{10}$ $R_{12,12}(r) = r^{12}$

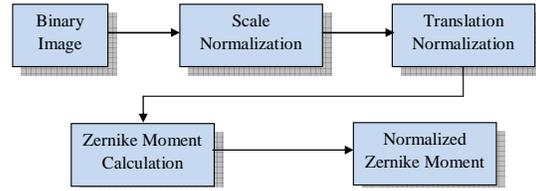


Figure 7. Zernike moment computations.

Scale normalization is done by using the following equation [21]

$$x' = x \sqrt{\frac{\beta}{m_{00}}}, \quad y' = y \sqrt{\frac{\beta}{m_{00}}},$$

where  $\sqrt{\frac{\beta}{m_{00}}}$  is a scale factor,  $\beta$  is a predetermined value ( $\beta = 20000$ ) and  $m_{00}$  is spatial moment order (0,0) that represents the area of a leaf.

Translation normalization is done by shifting the centroid of the leaf to the center of the image. In this case,  $x$  and  $y$  are computed by using

$$x_c = \frac{m_{10}}{m_{00}}, \quad y_c = \frac{m_{01}}{m_{00}}$$

In the above equation,  $(x_c, y_c)$  is the centroid,  $m_{10}$  is spatial moment order (1,0) and  $m_{01}$  is spatial moment order (0,1). Then, the new  $x$  and  $y$  can be calculated as

$$x' = x_c - \frac{N}{2}, \quad y' = y_c - \frac{M}{2},$$

where  $M$  is the height of the image and  $N$  is the width of the image.

Normalized Zernike moments are calculated by using

$$Z'_{pq} = \frac{Z_{pq}}{m_{00}}$$

## 2.2 Distance Measure in Identification System

CBIR (*content-based image retrieval*) is a system that can give a set of pictures based on a criterion. The mechanism of retrieving images can be based on query with a keyword, query by a sketch, query by example (QBE) and browsing by a category. In QBE, user can give an image, for

Based on Mukundan et al. [20], Zernike moments have the following advantages:

- rotation invariance: the magnitudes of Zernike moments are invariant to rotation;
- robustness: they are robust to noise and minor variations in shape;
- expressiveness: since the basis is orthogonal, they have minimum information redundancy.

However, Zernike moments depend on scaling and translation. So, in order to handle those problems, a procedure as shown in Figure 7 should be applied [17].

example a leaf, and the system tries to retrieve all images that have similarity with the query.

Similarity measurement between image of the query and the images in the database is done by measuring distance between feature vectors that represents two images. The distance measures that can be used are City block, Euclidean, Canberra, Bray-Curtis,  $\chi^2$  statistics, Kullback Liebler divergence and Jensen Shannon divergence [17][22]. However, generally, City block and Euclidean distances are commonly used because of its effectiveness and efficiency [17]. City block distance is defined as

$$d(Q, R) = \sum_{i=1}^N |Q_i - R_i|.$$

Euclidean distance is defines as

$$d(Q, R) = \sum_{i=1}^N (Q_i - R_i)^2$$

In both distance measures,  $d(Q,R)$  is distance between features in the query  $Q$  and features in the reference  $R$ . Meanwhile,  $N$  is number of features.

### 2.3 Probabilistic Neural Network

Probabilistic Neural Network is a well-known neural network that consists of 4 units: 1) input units, 2) pattern units, 3) summation units and 4) output units [23]. The architecture is shown in Figure 8. Input units receive value of features and then distribute them to every unit in the pattern units. The pattern units form a dot product of the input vector  $X$  with a weight vector  $W$  ( $W$  is generated by training data). So, the output of the unit  $i$  is  $Z_i$  where  $Z_i = X \cdot W_i$ . However, before sending the result to unit in the Summation units, a nonlinear operation on  $Z_i$  is done. The operation is  $\exp[(Z_i-1)/\sigma^2]$ . In this case,  $\sigma$  is the smoothing factor. In this research,  $\sigma = 0.1$ . The summation units sum the input from the pattern units. Then, the output units produce binary outputs that correspond to a class of the decision.

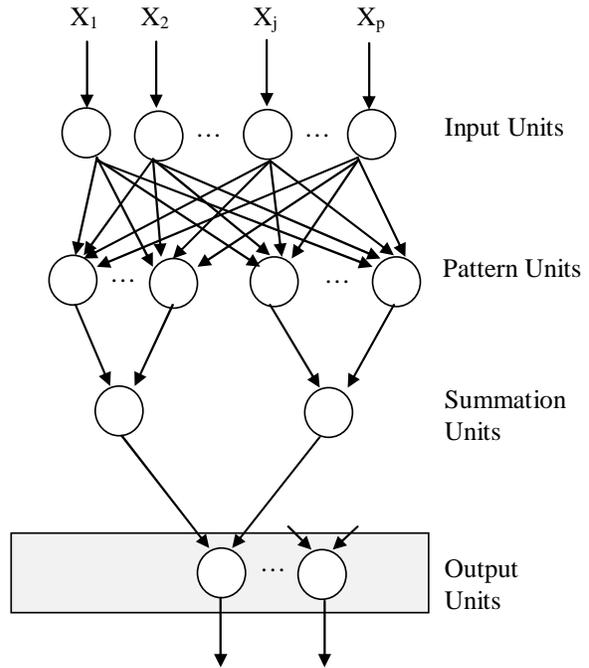


Figure 8. Architecture of PNN.

### 3. PROPOSED SYSTEMS

The first system was used a distance measure (City block distance or Euclidean distance) to determine the similarity index of the leaf of query to every leaf in the database. The smallest rank indicated the most similar one. However, in the experiment, five plants with the smallest ranks were selected as outputs. The mechanism to do the job is shown in Figure 9.

As shown in Figure 9, the leaf of the query is preprocessed and eventually the part of the leaf can be separated from its background. After that, similarity computations (using City block or Euclidean distance) between the leaf of the query and leaves in the database are accomplished. Based on the similarity indices produced from previous process, five different plants are obtained.

In order to measure the system accuracy, the classes of the five plants are compared to the class of query's leaf. Therefore, top 1, top 3 and top 5 can be calculated. In this case, the accuracy of the system was calculated by using

$$accuracy = \frac{n_r}{n_t}$$

where  $n_r$  is the relevant number of images and  $n_t$  is the total number of query.

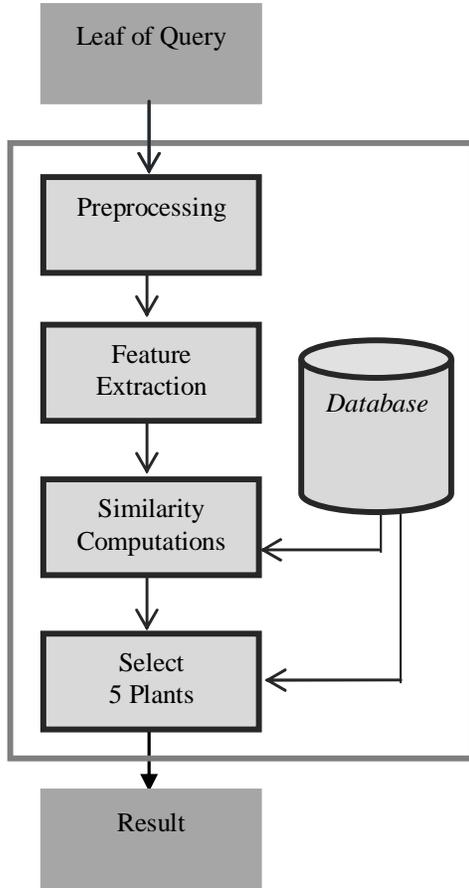


Figure 9. System by using similarity computations.

The second system used PNN. The block diagram of the system is shown in Figure 10. Same as the previous system, firstly, the leaf of the query are preprocessed to obtain the part of the leaf and then the features are extracted. Then, the features are processed by PNN to classify the leaf.

In the second system, the accuracy was also calculated by using

$$accuracy = \frac{n_r}{n_t}$$

where  $n_r$  is the relevant number of images and  $n_t$  is the total number of query.

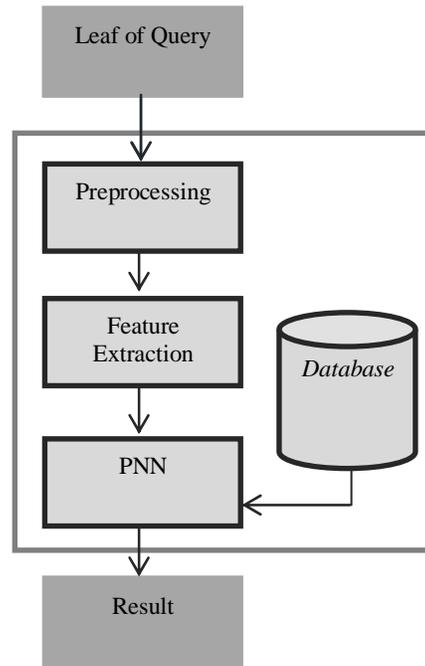


Figure 10. System using PNN.

#### 4. EXPERIMENTAL RESULTS

To test the both systems, two datasets were used. First dataset was Foliage, that contains 60 kinds of leaves with various colors and shapes. Second dataset was Flavia, that contains 32 kinds of leaves with green color.

##### 4.1 Testing using Dataset Foliage

The first system that used similarity computations was tested by involving 95 leaves per plant as references and 20 leaves per plant as testing data that came from dataset Foliage (that can be download at [rnd.akakom.ac.id/foliage/](http://rnd.akakom.ac.id/foliage/)). The system was tested by using City block distance and Euclidean distance separately. The results are shown in Table 2.

Based on Table 2, optimum accuracy 93.33% by using City block distance was reached when Zernike moments order 2 until order 7 were used. Using Euclidean distance, the optimum accuracy was reached when Zernike moments order 2 until order 12 were applied.



Table 2. Accuracy of the system using distance measures and dataset Foliage.

Order of Zernike Moments	Accuracy	
	City Block Distance	Euclidean Distance
2	90.25%	89.08%
2-3	90.08%	89.00%
2-4	90.67%	88.92%
2-5	91.75%	90.08%
2-6	92.83%	90.33%
2-7	93.33%	90.58%
2-8	93.08%	90.91%
2-9	92.25%	91.42%
2-10	91.50%	91.58%
2-11	91.25%	91.67%
2-12	90.58%	91.75%

Table 4. Accuracy of the system using distance measures and dataset Flavia.

Order of Zernike Moments	Accuracy	
	City Block Distance	Euclidean Distance
2	87.19%	90.31%
2-3	91.88%	92.50%
2-4	93.75%	94.06%
2-5	93.75%	93.44%
2-6	93.44%	93.75%
2-7	92.81%	94.06%
2-8	92.50%	94.69%
2-9	93.75%	93.44%
2-10	92.81%	94.37%
2-11	93.44%	94.37%
2-12	92.81%	94.69%

The PNN based system has been tested by involving 95 leaves per plant as references and 20 leaves per plant as testing data that came from dataset Foliage. The results are shown in Table 3. The optimum accuracy was reached when Zernike moments order 2 until order 12 were applied.

Table 3. Accuracy in PNN based system using dataset Foliage.

Order of Zernike Moments	Accuracy
2	90.42%
2-3	91.42%
2-4	91.50%
2-5	91.92%
2-6	91.50%
2-7	92.00%
2-8	91.92%
2-9	91.83%
2-10	91.50%
2-11	91.33%
2-12	92.92%

Based on Table 4, optimum accuracy 93.75% by using City block distance was reached when Zernike moments order 2 until order 4 were used. Using Euclidean distance, the optimum accuracy 94.69 was reached when Zernike moments order 2 until order 8 were applied.

The PNN based system has been tested by involving 40 leaves per plant as references and 10 leaves per plant as testing data that came from dataset Foliage. The results are shown in Table 5. The optimum accuracy was reached when Zernike moments order 2 until order 10 were applied.

Table 5. Accuracy in PNN based system using dataset Flavia.

Order of Zernike Moments	Accuracy
2	89.69%
2-3	92.50%
2-4	92.19%
2-5	92.50%
2-6	91.56%
2-7	90.93%
2-8	92.19%
2-9	92.50%
2-10	93.44%
2-11	92.50%
2-12	93.13%

#### 4.2 Testing using Dataset Flavia

The system that used similarity computations was also tested by using dataset Flavia, by involving 40 leaves per plant as references and 10 leaves per plant as testing data. The system was tested by using City block distance and Euclidean distance separately. The results are shown in Table 4.

### 4.3 Comparison with Other Systems

To compare to other systems, several researches that used Flavia and their results are listed in Table 6. Based on the table, the proposed systems that used Zernike moments gave promising results for plant identification.

Table 2. Accuracy comparison.

Methods	Accuracy	Note
PNN in [1]	90.31%	32 classes
PNN-PCNN in [8]	91.25%	32 classes
Fourier Moment in [8]	46.30%	32 classes
LDA in [24]	94.30%	20 classes
Proposed systems:		32 classes
• Euclidean distance	94.69%	
• City block distance	93.75%	
• PNN	93.44%	

## 5. CONCLUSIONS

Based on the experiments, Zernike moment have a prospect to be included in the leaf identification. The important thing should be considered is how many moments should be included in the system in order to achieve the highest performance.

The experiments also show that neither City block nor Euclidean distance is superior to the other when system used a distance measure to decide decision in identifying leaf. Therefore, testing the system by using each distance measure can help in finding the optimum accuracy of the leaf identification system.

### REFERENCES:

- [1] S.G. Wu, F.S. Bao, E.Y. Xu, Y.X. Wang, Y.F. Chang, and Q.L. Xiang, (2007). "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network", IEEE 7th International Symposium on Signal Processing and Information Technology, 2007, Cairo.
- [2] Z. Zulkifli, "Plant Leaf Identification Using Moment Invariants & General Regression Neural Network", Master Thesis, Universiti Teknologi Malaysia, 2009.
- [3] Q.K. Man, C.H. Zheng, X.F. Wang, and F.Y. Lin, "Recognition of Plant Leaves Using Support Vector", *International Conference on Intelligent Computing*, 2008, pp. 192-199, Shanghai.
- [4] J. Chaki, and R. Parekh, "Designing an Automated System for Plant Leaf Recognition", *International Journal of Advances in Engineering & Technology*, Vol. 2, Issue 1, Januari 2012, pp. 149-158.
- [5] Q. Wu, C. Zhou, and C. Wang, "Feature Extraction and Automatic Recognition of Plant Leaf using Artificial Neural Network", *Avances en Ciencias de la Computacion*, pp. 5-12.
- [6] A. Ehsanirad, and Sharath Kumar, "Leaf recognition for Plant Classification Using GLCM and PCA Methods", *Oriental Journal of Computer & Science & Technology*, Vol. 3, No. 1, 2010, pp. 31-36.
- [7] L. Jiming, "A New Plant Leaf Classification Method based on Neighborhood Rough Set", *Advances in information Sciences and Service Sciences(AISS)*, Vol. 4, No. 1, January 2012, pp. 116-123.
- [8] K. Singh, I. Gupta, and S. Gupta, "SVM-BDT PNN and Fourier Moment Technique for Classification of Leaf Shape". *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 3, No. 4, 2010, pp. 67-78.
- [9] C.L. Lee, and S.Y. Chen, "", 16<sup>th</sup> IPPR Conference on Computer Vision, Graphics and mage Processing (CVGIP), 2003, pp. 355-362.
- [10] M.S. Nixon and A.S. Aguado, "Feature Extraction and Image Processing", Newnes, 2002.
- [11] J.C. Russ, "The Image Processing Handbook". CRC Press, Boca Raton, 2011.
- [12] M.T. Goodrich and R. Tammaia, "Algorithm Design", John Wiley & Sons, 2002.
- [13] W. Martinez and A. Martinez, "Computational Statistics Handbook With MATLA", CRC Press LLC, Bocca Raton, 2002.
- [14] R. Dobrescu, M. Dobrescu, S. Mocanu, and D. Popescu, "Medical Images Classification for Skin Cancer Diagnosis Based on Combined Texture and Fractal Analysis", *WISEAS Transactions on Biology and Biomedicine*, Vol 7, No. 3, 2010, pp. 223-232.
- [15] H.B. Kekre, S.D. Thepade, T.K. Sarode, V. Suryawanshi, "Image Retrieval using Texture Features extracted from GLCM, LBG and KPE", *International Journal of Computer Theory and Engineering*, Vol. 2, No. 5, October, 2010, 695-700.



- [16] T. Acharya and A. Ray, "Image Processing Principles and Applications", John Wiley & Sons, Inc, New Jersey, 2005.
- [17] D. Zhang, "Image Retrieval Based on Shape", Unpublished Dissertation, Monash University, 2002.
- [18] J. Flusser, T. Suk and B. Zitova, "Moments and Moment Invariants in Pattern Recognition", John Wiley & Sons, Ltd, Chicester, 2009.
- [19] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, and S. Verzakov, "PRTools 4.1, A Matlab Toolbox for Pattern Recognition", Delft University of Technology, 2007.
- [20] Y. Mingqiang, K. Kidiyo, R. Joseph, "A Survey of Shape Feature Extraction Techniques", Pattern Recognition, Peng-Yeng Yin (Ed.), 2008, pp. 43-90.
- [21] S. Li, M.C. Lee, C.M. Pun, "Complex Zernike Moments Features for Shape-Based Image Retrieval", IEEE Transactions on Systems, Man, and Cybernetics, Part A: System and Humans, Vol. 39, No. 1, 2009, pp. 227-237.
- [22] T. Deselaers. 2003. *Features for Image Retrieval*. [Online]  
[http://thomas.deselaers.de/publications/papers/deselaers\\_diploma03.pdf](http://thomas.deselaers.de/publications/papers/deselaers_diploma03.pdf). Accessed on February 10, 2012.
- [23] D.F. Specht, "Probabilistic Neural Networks", Neural Networks, 1990, Vol. 3, pp. 109-118.
- [24] M. Shabanzade, M. Zahedi, and S.A. Aghvami, "Combination of Local Description and Global Features for Leaf Recognition", Signal & Image Processing: An International Journal, Vol. 2, No. 3, 2011, pp.23-31.