British Journal of Mathematics & Computer Science
4(15): 2217-2236, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

Zernike Moments and Genetic Algorithm : Tutorial and Application

Babatunde Oluleye[1*], Armstrong Leisa[1], Jinsong Leng[1] and Diepeveen Dean[2]

[1] *eAgriculture Research Group, School of Computer and Security Science, Edith Cowan University, Perth, Western Australia, Australia.*

[2] *Department of Agriculture and Food, Perth, Western Australia, Australia.*

Original Research Article

Received: 16 April 2014
Accepted: 27 May 2014
Published: 12 June 2014

Abstract

Aims/ objectives: To demontrate effectiveness of Zernike Moments for Image Classification.
Zernike moment(ZM) is an excellent region-based moment which has attracted the attentions of many image processing researchers since its first application to image analysis. Many papers have been published on several works done on ZM but no single paper ever give a detailed information of how the computation of ZM is done from the time the image is captured to the computation of ZM. This work showed how to effectively apply ZM on RGB images. We have demonstrated the effectiveness of Zernike moment in image classification system. A neuro-genetic intelligent system has been built with PNN classifier. The feature extracted viz ZM and Geometric features were further subjected to GA to bring the best combinatorial features for optimal accuracy. The algebraic structure of our novel fitness function enabled the GA to select the best results. The 10-fold CV used enabled the whole system to be unbiased giving a classification accuracy of 90.05%. A demonstration of affine properties of ZM are comprehensively stated and explained. In summary, the ZM enabled the classifier to have improved accuracy of 91% as compared with Geometric features with 89 % accuracy.

Keywords: Zernike Moment ;Feature Extraction, Geometric Properties, Image Classification
2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

1 Introduction

In object/image recognition, a region can be described using scalar or set of scalars based on the geometric properties of the object. Such scalars or set of scalars are called descriptors since they descripe objects being recognised by artificial vision system. This work presents a novel and detailed framework for efficient computation of Zernike Moment (ZM) - a region-based moments.

**Corresponding author: E-mail: hezecomp@yahoo.com*

Using ZM and geometric features, we extracted 20 features, which were later subjected to a Genetic Algorithm (GA) for dimensionality reduction to obtain best feature set which we then used to build our classification system.

2 Image Moments

A moment describe the layout (arrangement of image pixels). Moments are global region-based descriptors for shape and is bit like combination of area, compactness, irregularity, and higher order descriptors together [1]. An image moment is defined as the integration of an image function with a region-defined polynomial basis ([2,3]). The region here is defined as the area where that image is valid. From [4], the general moment M_{pq} of any image $f(x,y)$ of order $p+q$, where $p>0, q>0$, is defined as:

$$M_{pq} = \int\int_D pol_{pq}(x,y)f(x,y)dxdy \tag{2.1}$$

where $pol_{pq}(x,y), i = 1(0)p, j = 1(1)q$ are polynomials basis functions defined on domain D.

3 Zernike Moment

The Zernike moment(ZM) can be defined as a set of complete complex orthogonal basis functions that are square integrable and that are defined over the unit disk. ZM were first applied in image analysis for the first time in [5]. ZM are orthogonal moment based on Zernike polynomials (see Table 1). Orthogonality here means that there is no redudancy or overlapping of information between the moments. Thus moments are uniquely quantified based on their orders ([6,7]). The distinguishing feature of ZM is the invariance of its magnitude with respect to rotation ([8,9,10,11]). If we are given the ordered pair (m,n) which represents the order of the Zernike polynomial and the multiplicity of its phase angle, then the ZM, can be defined as

$$V_{nm}(\rho,\theta) = R_{nm}(\rho)e^{im\theta}, \theta \leq 1 \tag{3.1}$$

where

$$\rho = \sqrt{x^2+y^2}, \theta = \arctan(\frac{y}{x}) \tag{3.2}$$

are the image pixel radial vector and angle between it and x-axis respectively

$$R_{nm}(\rho) = \sum_{a=0}^{\frac{(n-|m|)}{2}} (-1)^a \frac{(n-a)!}{a!(\frac{(n+|m|)}{2}-a)!(\frac{(n-|m|)}{2}-a)!}\rho^{n-2a} \tag{3.3}$$

The R_{nm} is the Zernike/radial basis polynomial, some of which are listed in Table 1. The following conditions must be satisfied:

$$(a) n \in Z^+$$
$$(b) n-|m| \quad is \;\; even$$
$$(c) |m| \leq n$$
$$(d) \int_0^{2\pi}\int_0^1 V_{nm}^*(\rho,\theta)\rho d\rho d\theta = \frac{\pi}{n+1}\delta_{np}\delta_{mq}, \delta_{zv} = \begin{cases} 1 & z=v, \\ 0, & otherwise \end{cases} \tag{3.4}$$

Table 1: Zernike Polynomials

S/N	Representation	Formular
1	$R_{0,0}(r)$	1
2	$R_{1,1}(r)$	r
3	$R_{2,0}(r)$	$2r^2 - 1$
4	$R_{2,2}(r)$	r^2
5	$R_{3,1}(r)$	$3r^3 - 2r$
6	$R_{3,3}(r)$	r^3
7	$R_{4,0}(r)$	$6r^4 - 6r^2 + 1$
8	$R_{4,2}(r)$	$4r^4 - 3r^2$
9	$R_{4,4}(r)$	r^4
10	$R_{5,1}(r)$	$10r^5 - 12r^3 + 3r$
11	$R_{5,3}(r)$	$5r^5 - 4r^3$
12	$R_{5,5}(r)$	r^5
13	$R_{6,0}(r)$	$20r^6 - 30r^4 + 12r^2 - 1$
14	$R_{6,2}(r)$	$15r^6 - 20r^4 + 6r^2$
15	$R_{6,4}(r)$	$6r^6 - 5r^4$
16	$R_{6,6}(r)$	r^6
17	$R_{7,1}(r)$	$35r^7 - 60r^5 + 30r^3 - 4r$
18	$R_{7,3}(r)$	$21r^7 - 30r^5 + 10r^3$
19	$R_{7,5}(r)$	$7r^7 - 6r^53$
20	$R_{7,7}(r)$	r^7
21	$R_{8,0}(r)$	$70r^8 - 140r^6 + 90r^4 - 20r^2 + 1$
22	$R_{8,2}(r)$	$56r^8 - 105r^6 + 60r^4 - 10r^2$
23	$R_{9,1}(r)$	$126r^9 - 280r^7 + 210r^5 - 60r^3 + 5r$
24	$R_{9,3}(r)$	$84r^9 - 168r^7 - 105r^5 - 20r^3$
25	$R_{9,5}(r)$	$36r^9 - 56r^7 + 21r^5$
26	$R_{9,7}(r)$	$9r^9 - 8r^7$
27	$R_{9,9}(r)$	r^9

4 Computation of Zernike Moments for Images of Plant leaves

Given the definitions in Equations 3.1-3.4, the Zernike moment, Z_{nm} for an image $\{f(x_i, y_j) : 1 \leq i \leq M, 1 \leq j \leq N\}$, can be calculated as Equations (4.1) or (4.2)

$$Z_{nm} = \frac{n+1}{\pi} \int\int_D f(x,y) V^*_{nm}(x,y) dxdy = \frac{n+1}{\pi} \sum_x^M \sum_y^N V^*_{nm}(x,y) f(x,y) \tag{4.1}$$

where $x^2 + y^2 \leq 1$, and $m = 0, 1, 2, 3, ...\infty$. The m defines the order of the Zernike Polynomial while n which is either negative or positive, represents the multiplicity of the phase angles in ZM.

$$Z_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta) R_{nm}(\rho) e^{-im\theta} \rho d\rho d\theta \tag{4.2}$$

4.1 Image Pre-Processing Steps

Input: The original RGB image of a plant leaf (1) is given and can also be represented as $\{f(x_i, y_j) : 1 \leq i \leq M, 1 \leq j \leq N\}$. The dimension of the RGB leaf image in figure (1) is $1200 \times 1600 \times 3$

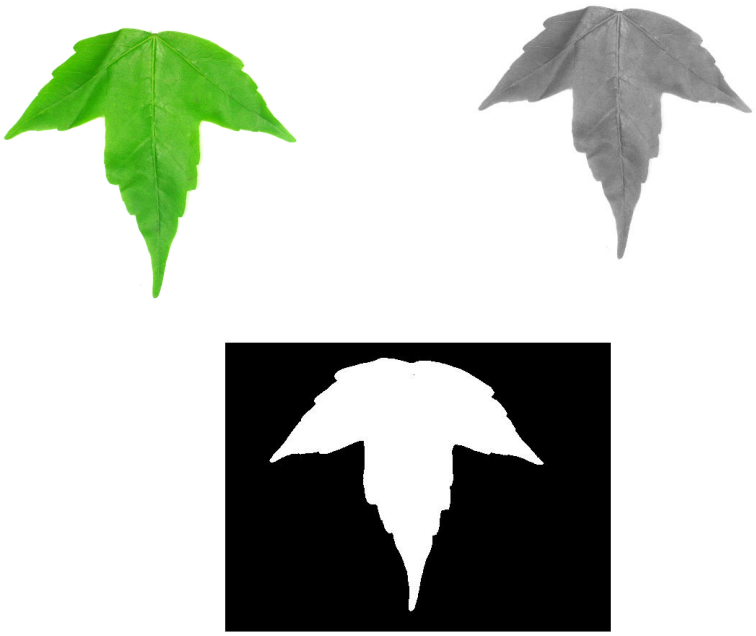

Figure 1: RGB, Grayscale, and Binary version of a plant's leaf

containing different integers $\{Z_i, i = 1(1)3 : 1 \leq Z_i \leq 255\}$, representing **RGB** value for the leaf's image. After reading the RGB image from figure (1) into MATLAB workspace, then we have the following numbers as output (represented as SET A)

SET A = [76 75 77 82 88 93 95 93 79 76 73 71 71 75 81 84 81 83 85 86 83 81 83 85 94 109 117 113 105 99 87 75 79 80 83 88 91 90 87 82 74 71 68 66 66 70 74 77 68 73 79 83 81 77 74 76 101 112 118 110 101 93 82 7281 82 87 92 93 87 78 69 68 66 64 62 62 64 67 69 63 72 82 90 89 101 100 98 ...255].
Therefore n(SET B) = 1200 * 1600 * 3 = 5760000. In order words, n represents the total numbers used to represent the R,G,B values from the image.

STEP1: Conversion to grayscale
The RGB image is converted to grayscale using appropriate formula. A good example (which is used here) is the $rgb2gray()$ function from MATLAB and is expressible as equation (4.3)

$$Grayscale = 0.2989R + 0.5870G + 0.1140B \tag{4.3}$$

where R, G, and B correspond to the red, green and blue colour of the pixel, respectively, while their coefficients represent human perception of the colour. According to [12], the retina of human eyes

has three types of cone which are L-cone (most sensitive to red light), M-cone (most sensitive to green light), and S-cone (most sensitive to blue light). The formula in Equation 4.3 was designed to match human brightness perception based on M-cone of human eyes retina. The simplest colour-to-grayscale algorithm, which is the mean of the RGB channels, is given in Equation 4.4.

$$Grayscale = \frac{R+G+B}{3} = 0.3333R + 0.3333G + 0.3333B \tag{4.4}$$

The size of the grayscale image in figure 1 is 1200×1600. Thus, the total number of pixels here is 1200 x 1600 = 1920000. This implies the pixel values (intensity information) are spread across a rectangular or square region. This is a monochrome image. The numerical output for this image is $\{Z_i\}_{i=1}^{1920000} : 1 \leq Z_i \leq 255^{\cdot}$
Some of the pixel values are listed (as SET B) below:

SET B = [255 255 255 255 255 255 255 255 255 255 ...19 116 126 125 122 120 119 120 120 121 130 133 140 148 154 159 254 253 251 247 245 242 239 234 226 217 210 188 173 152 138 134 134 135 135 145 147 148 148.255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255...].
Similarly, n(SET B) = 1200 * 1600 = 1920000. In order words, n represents the total numbers used to represent total number of intensities in the grayscale image of the leaf.

STEP 2: Conversion to Binary Image
This steps further simplifies the greyscale image from STEP 2 to binary image to facilitate further computation. A good function to realize this conversion is im2bw() from MATLAB Toolbox. A thresholding such as otsu method is commonly used in MATLAB to achieve this (after grayscale conversion).Figure 1c represents a typical output from this step. The size of the binary image is still 1200×1600. This implies the total number of pixels information here is still 1200 x 1600 = 1920000. The numerical output for this image is$\{Z_i, i = 1 : Z_i \in \{0,1\}\}$. Some of the pixel values are listed (as SET C) below:

SET C = [...1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...].

The dimension of SET B = dimension of SET C but the contents of SET C are bits which are easier for feature extractors and even computers to process.

5 Computational Algorithms and Image Normalization

PROBLEM STATEMENT: The problem here is to slide pixels information for the binary image from rectangular or square grid over a unit disk as shown in figure 2.

5.1 Options for Computation of ZM

The ZM can be computed using
(a)Cartesian Coordinate system and
(b)Polar Coordinate System

•Cartesian Coordinate System
Given a digital image $f(x_i, y_j)$, $, i = 1(1)N$, $, j = 1(1)N$. The xy-space is discretised as $x_i = (2i - N - 1)/N$ and $y_j = 2j - N - 1)/N$, where the double summation is performed over the (i, j)

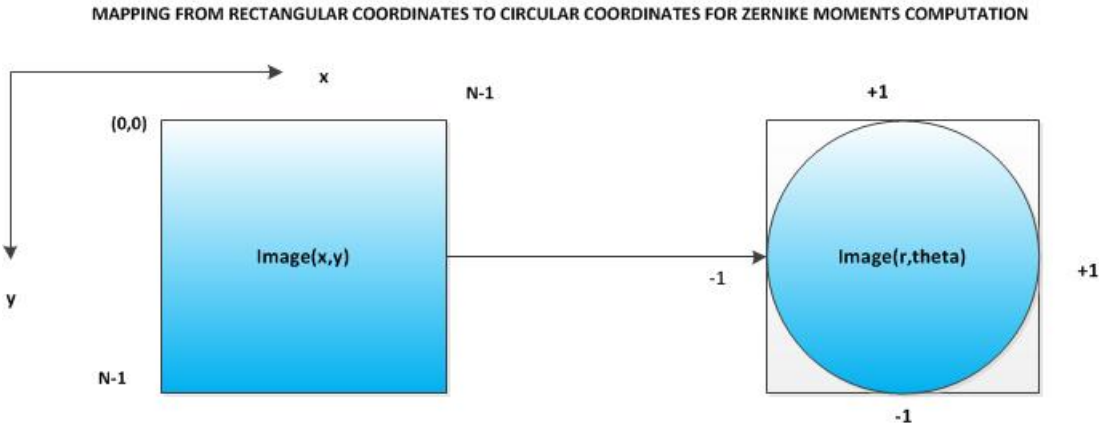

Figure 2: Conversion from rectangular to polar coordinates

pairs that satisfy Equation 4.1. The function represents the image being described (in this case ROI (Fig. 1(c)) from a plant leaf image). Integer m is either positive or negative, depicting the angular dependence or rotation, subject to the conditions (b) and (c) of Eq 3. The asterisk over the function V means complex conjugate.

•Polar Coordinate System
The ROI (1(c) is mapped to the unit disc (using Eq 3.2) through polar coordinates, where the center of the ROI is the origin of the unit disk. The conversion from rectangular to polar coordinates is done through Eq (3.2). The coordinates are then described by the length of the vector from the origin of the disk to the coordinate point ρ , and the angle from the x-axis, to the vector ρ, (the polar radius). The polar angle is represented as θ. The pixels falling outside the unit disc are not used in the calculation. The translation invariance is achieved by moving the centroid of the ROI to the origin of the disk and this eventually causes $m_{01} = m_{10} = 0$. The centroid of the ROI is given by the coordinates $(\bar{x}, \bar{y})$ where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \tag{5.1}$$

The scale invariance for ZM is achieved through normalization of the image so that the total area of the forground pixels is of predetermined value, say, β.
Translation and scaling invariance is then achieved through the formula in Eq (5.2).

$$g(x, y) = f\big(\frac{x}{a} + \bar{x}, \frac{y}{a} + \bar{y}\big) \tag{5.2}$$

where

$$a = \sqrt{\frac{\beta}{m_{00}}} \tag{5.3}$$

The results of our computation showed that the numerical values of the ZM with respect to tranlation, rotation, and scalings are invariant or of negligble differences. The numerical values for the ZM computation are shown in Table (2). The rotations, scalings, and translation were taken over $\{15, 35, 45, 60\}$, $\{0.25, 0.5, 0.75\}$, and $\{(1,2)\}$, repectively. The results in Table (2) represent the ZM values computed on the first twenty set of leaves taken from the Flavia dataset [13]. Figure (3) showed the results of ZM computation over original ROI, rotated ROI at angle 30, scaled ROI (by 0.75) and translated ROI (at (1,2)). Similarly, Figures (6(a) and (b) showed the graph of ZM aplitude plotted

against angles of rotattion and scaling values respectively. This two plots being parrallel to x-axis also mean that the computation of ZM of the same order and repetition across different angles, scalings and tranlated of the original ROI is remain constant. A re-constructed version of the original binary image of a leaf are shown (see Figure 7) across different orders (orders 5 to 60) of ZM showing the correctness of ZM computation in this work. The figure shows that order 60 is sufficient to reconstruct original image.

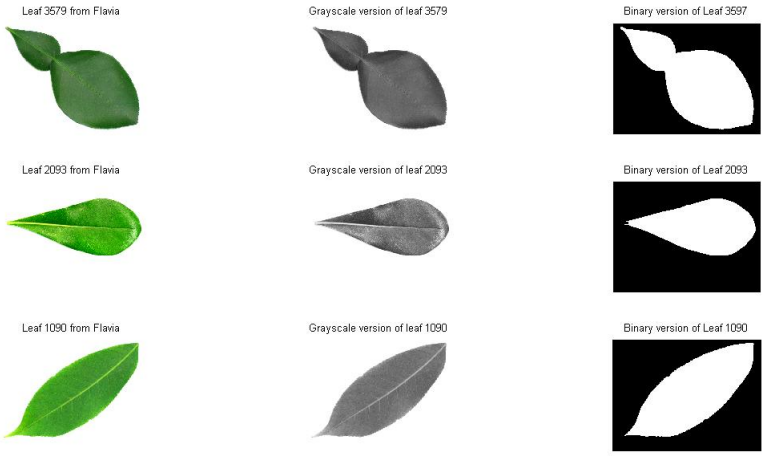

Figure 3: Leaves of three species of plant taken from the Flavia dataset

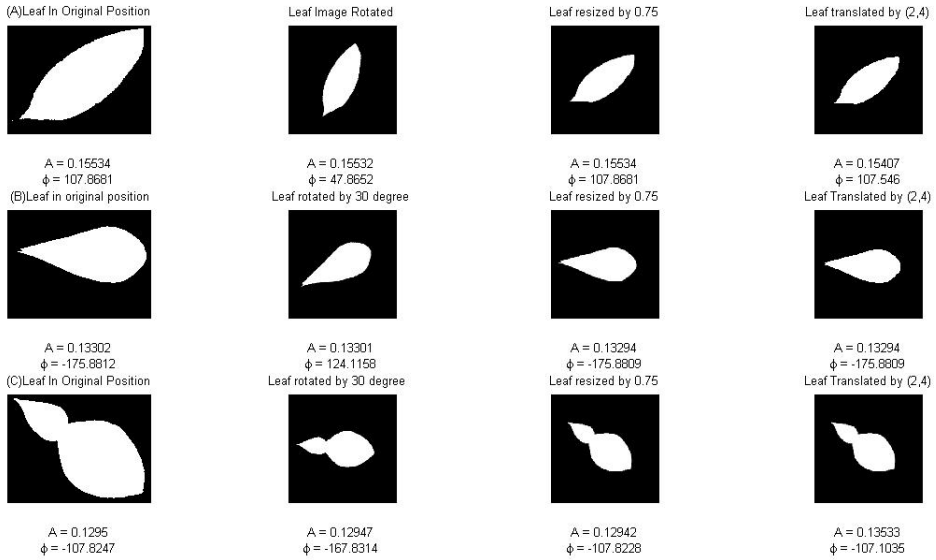

Figure 4: Invariance property of ZM under Translation, Rotation, and Scalings

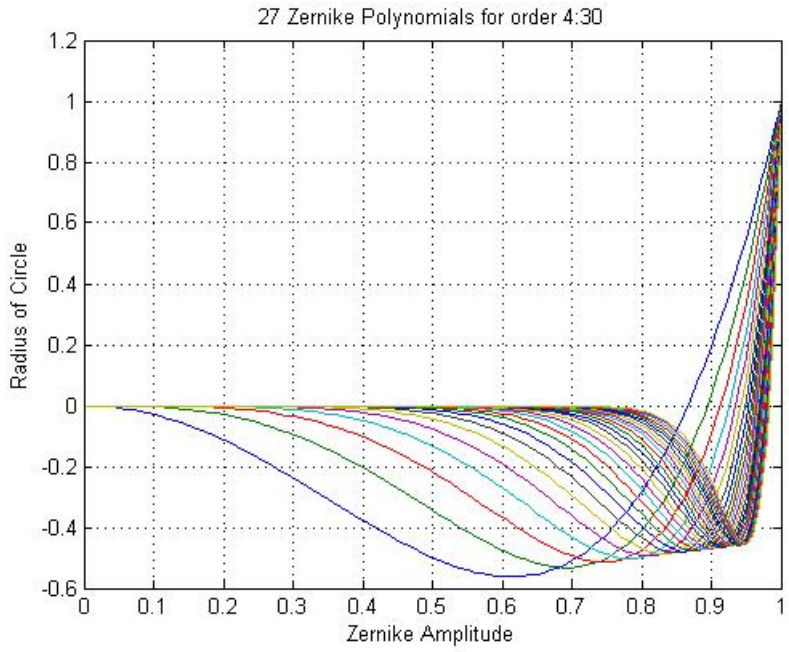

Figure 5: Zernike Polynomials for orders 4 to 30

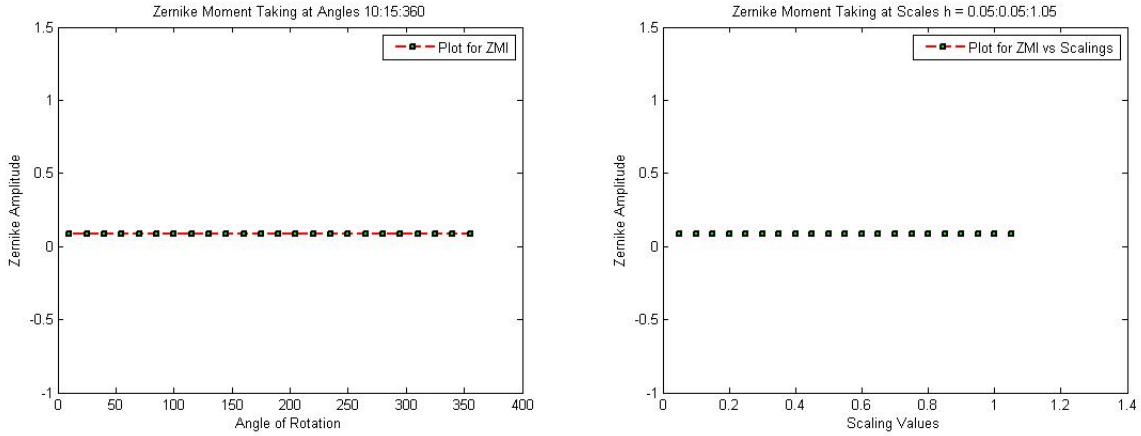

Figure 6: Zernike moment amplitude plot against angles and scalings

Table 2: Invariant ZM under Translation, Rotation, and Scaling

No	ZMI	ZMI15D	ZMI35D	ZMI45D	ZMI60D	ZMI0.25	ZMI0.35	ZMI0.5	ZMI0.75	ZMIT12
1	0.057920	0.057914	0.057922	0.058099	0.057912	0.057702	0.056961	0.057805	0.057901	0.057397
2	0.056176	0.056170	0.056169	0.056326	0.056170	0.055964	0.055240	0.056064	0.056145	0.055734
3	0.058929	0.058922	0.058923	0.058641	0.058923	0.058707	0.057920	0.058811	0.058896	0.058430
4	0.059335	0.059329	0.059329	0.059102	0.059329	0.059111	0.058309	0.059216	0.059267	0.058829
5	0.053977	0.053973	0.053969	0.053935	0.053983	0.053773	0.052937	0.053869	0.053942	0.053797
6	0.060549	0.060541	0.060539	0.060529	0.060544	0.060320	0.059404	0.060428	0.060517	0.060487
7	0.061169	0.061161	0.061164	0.061230	0.061157	0.060938	0.060051	0.061046	0.061115	0.061008
8	0.060348	0.060340	0.060343	0.060171	0.060342	0.060120	0.059264	0.060227	0.060294	0.060128
9	0.054818	0.054812	0.054800	0.054805	0.054812	0.054611	0.053941	0.054709	0.054776	0.054433
10	0.056890	0.056880	0.056871	0.056928	0.056889	0.056675	0.055984	0.056776	0.056862	0.056493
11	0.062296	0.062289	0.062299	0.061947	0.062296	0.062061	0.061335	0.062171	0.062243	0.061848
12	0.061393	0.061386	0.061380	0.061467	0.061387	0.061162	0.060266	0.061271	0.061347	0.061276
13	0.055376	0.055370	0.055357	0.055615	0.055377	0.055168	0.054555	0.055266	0.055341	0.054934
14	0.060846	0.060840	0.060838	0.060906	0.060837	0.060617	0.059757	0.060725	0.060792	0.060655
15	0.063584	0.063578	0.063583	0.063392	0.063578	0.063344	0.062522	0.063457	0.063552	0.063196
16	0.063465	0.063456	0.063454	0.063408	0.063454	0.063226	0.062355	0.063339	0.063417	0.063186

17	0.059430	0.059422	0.059422	0.059403	0.059422	0.059206	0.058365	0.059311	0.059399	0.059222
18	0.062665	0.062659	0.062653	0.062866	0.062650	0.062429	0.061587	0.062540	0.062626	0.062381
19	0.058247	0.058240	0.058237	0.058220	0.058237	0.058028	0.057243	0.058131	0.058240	0.057891
20	0.057960	0.057953	0.057947	0.057930	0.057963	0.057742	0.056939	0.057844	0.057953	0.057617

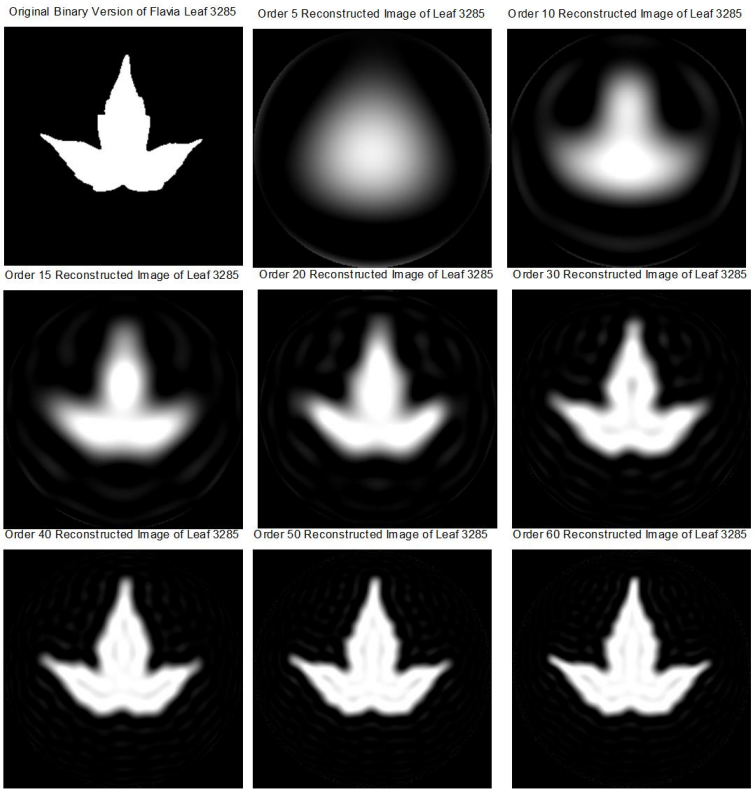

Figure 7: Original and Reconstructed Image

6 Geometric Features

We extracted the following geometric features from the Flavia dataset:

- **Diameter**: This is the longest distance between any two coordinates on the margin of a leaf.
- **Physiological Length**: This is the distance between the two terminals (apex and stalk point) [14].

- **Physiological Width**: This is the perpendicular distance across the physilogical length of a leaf [13].
- **Leaf Area**: This is the total number of pixels that constitute an image $a = \int_x \int_y I(x,y)dydx$.
- **Aspect Ratio**: This is also called eccentricity and is defined as ratio between length of the leaf minor axis and the length of the leaf major axis [15]. $\frac{w}{l}$
- **Circularity**: This is a measure of similarity between a 2D shape is and a circle. It is the ratio between area a of the leaf and the square of its perimeter p . It is given as $\frac{a}{p^2}$. [14].
- **Irregularity**: This is the ratio between the radius of the maximum circle encompassing the region and the minimum circle that can be contained in the region ([16,17]) . It is given as $\frac{max(\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2})}{min(\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2})}$
- **Solidity**: This is defined as the ratio between the area of the leaf and the area of its convex hull [14]. It is given as $\frac{a}{AreaOfConvexHull}$.
- **Form Factor**: This feature describes the difference between a leaf and a circle. It is represented as $\frac{4\pi a}{p}$, where a is the leaf area and p is the perimeter of the leaf.
- **Rectangularity**: This describes the similarity between a leaf and a rectangle. It is represented by $\frac{lw}{a}$ where l is the physiological length, w is the physiological width and a is the leafs area.

7 Feature Space and Feature Selection

We extracted 20 features from the Flavia dataset. The Flavia dataset comprises of 1907 colored images of 32 species of plants [13]. These features are (a) 10 Zernike moments and (b) 10 geometric moments. Thus the dimensionality of the dataset is 1907×20. High dimensional feature set could pose a great threat to pattern or image recognition systems. In otherwords, too many features sometimes reduce the classification accuracy of the recognition system since some of the features may be redundant and non-informative [18]. Different combinatorial set of features should be obtained in order to keep the best combination to achieve optimal accuracy. As such, a GA-based feature selection (a subspace or manifold projection technique) will be used to reduce the number of features needed by the PNN Classifier in this work. A Feature Subset Selection (FSS) is an operator Fs or a map from an **m-dimensional feature** space (input space) to **n-dimensional** feature space (output) given in mapping,

$$Fs : R^{r\times m} \mapsto R^{r\times n} \tag{7.1}$$

where $m \geq n$ and $m, n \in Z^+$, $R^{r\times m}$ is any database or matrix containing the original feature set having r instances or observation, $R^{r\times n}$ is the reduced feature set containing r observations in the subset selection.

7.1 Feature Selection Using Genetic Algorithm

Genetic Algorithms (GA) can be defined as population-based and algorithmic search heuristic methods that mimic natural evolution process of man ([19,20]). GA iteratively employ the use of one population of chromosomes (solution candidates) to get a new population using a method of natural selection combined with genetic functionals such as crossover and mutation (in the similitude of Charles Darwin evolution principle of reproduction, genetic recombination, and the survival of the fittest). In comparative terminology to human genetics, chromosomes are the bit strings, gene is the feature, allele is the feature value, locus is the bit position, genotype is the encoded string, and phenotype is the decoded genotype [21]. The fitnesses of the chromosomes are evaluated using a function commonly refered to as **Objective function or fitness function**. In other words, the fitness function

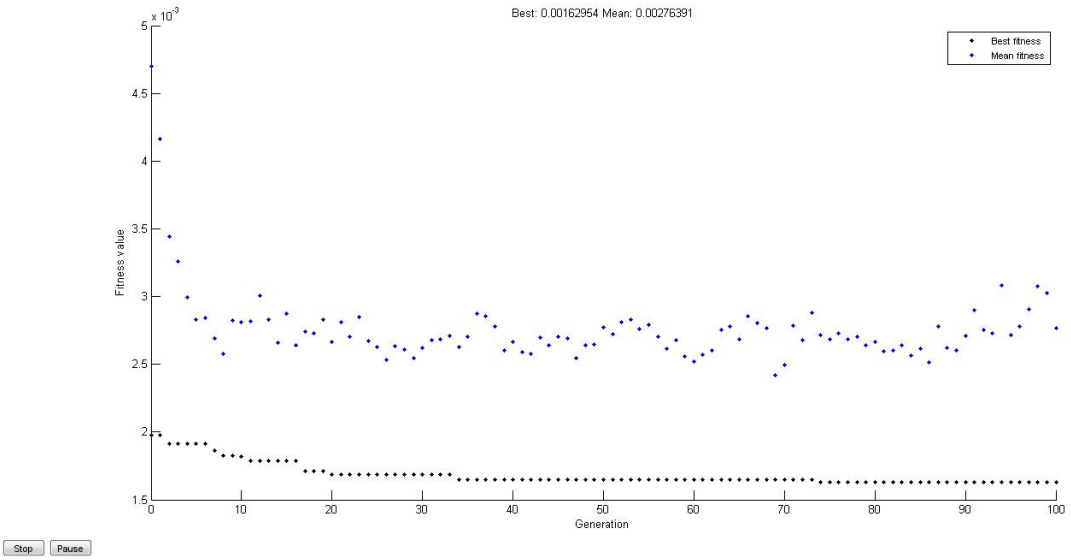

Figure 8: Convergence of GA Algorithm

(objective function) reports numerical values which are used in ranking the chromosomes in the population. Thus, the five important issues in the GA are chromosome encoding, population initialization, fitness evaluation, selection (followed by genetic operators), and criteria to stop the GA (see Figure 9). The GA operates on binary search space as the chromosomes are bit strings. The GA manipulates the finite binary population in similitude of human natural evolution. First, an initial population is created randomly and evaluated using a fitness function. As regards binary chromosome used in this work, a gene value **'1'** indicates the particular feature indexed by the position of the **'1'** is selected. If it is **'0'**, the feature is not selected for evaluation of the chromosome concerned. The chromosomes are then ranked and based on the rankings, the top n fittest kids (Elitism of size n) are selected to survive to the next generation. The fitness evaluation is done through Algorithm 7.1. The fitness function used in 7.1 is shown in Equation 7.2. After the elite individuals are moved to the next generation, the remainning individuals in the current population are used to produce the rest of the next generation through crossover and mutation. Crossover is basically, combination of two individuals to form a crossover kid. Mutation operator on the other hand, is a genetic pertubation of the genes in each chromosomes through flipping of bits depending on the mutation probability. The configuration for our GA is shown in Table 3. The surviving chromosome for GA is the string BestChromosome = $\{0$ 0 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 $0\}$. The positional indices of "1s" in this string are $\{3$ 6 8 11 12 14 $17\}$. The corresponding features from this string are those features positions $\{3$ 6 8 11 12 14 $17\}$. These features are Zernike moments with the following orders and repetitions {(2,0), (4,2),(5,3) }, eccentricity, form factor, EulerNumber, and Leaf minor axis. These features are used to train the PNN classifier used in Figure 11.

$$FitFunc1 = \frac{\alpha}{N_f} + \exp\left(-\frac{1}{N_f}\right) \tag{7.2}$$

where
α = kNN-Based classification error and N_f = Cardinality of the selected features. The algebraic structure of Equation 7.2 ensures the learning of the GA, error minimization and reduced number of features selected.

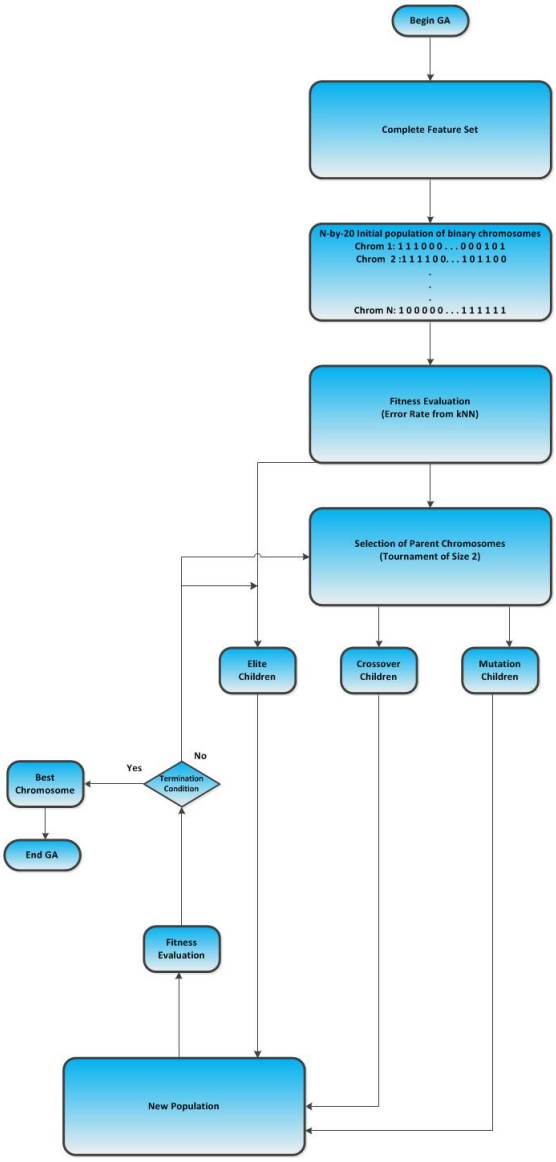

Figure 9: GA-Based Feature Selection

Algorithm 7.1. *Fitness Function Evaluation*

1: ***procedure*** FITFUNCTION1()
2: *FeatIndex* ← *Indices of ones from BinaryChromosome*
3: *NewDataSet* ← *DataSet indexed by FeatIndex*
4: *NumFeat* ← *Number of elements in FeatIndex*
5: *3* ← *NumNeighborskNN*

Table 3: Parameters Used in GA

GA Parameter	**Value**
Population size	100
Genomelength	20
Population type	bitstrings
Fitness Function	kNN
Number of generations	100
Crossover	Arithmetic Crossover
Crossover Probability	0.8
Mutation	Uniform Mutation
Mutation Probability	0.1
Selection scheme	Tournament of size 2
EliteCount	2

6: $kNN_{Error} \leftarrow Classifier_{KNN}$*(DataSet,ClassInformation,NumNeighborskNN)*
7: *Return* kNN_{Error}
8: ***end procedure***

8 Image Classification System

Our classification system shown in Figure 11 is based on Probabilistic Neural Network (PNN) which is a feed forward Neural Network that uses kernel methods for density estimation in a multi-category problem and which was introduced by D.F Specht [22]. PNN can be seen as a mathematical interpolation [23] or a parallel implementation of Parzen type classifier model. The algorithmic description of our classifier is shown in Algorithm 8.1. The whole system was built completely from MATLAB version 2013. The PNN spread was chosen to be in the neighborhood of $\frac{1}{n}$, where n = number of classes (in this case 32). This was a proof by GA-Optimization technique which is not part of the scope of the current paper. So the over fitting of the PNN classifier was properly checked.

Algorithm 8.1. *Pseudocode for PNN Classifier*

1: ***procedure*** PNNCLASSIFY(TRAININGSET, TESTSET, SPREAD)
2: ***Input TrainingSet*** *and the class information.*
3: *Compute $N_i, p(c_i)$, where N_i = Number of training patterns in each class c_i = class information, i = 1(1)32.*
4: *Initialize PNN Spread as $\sigma \mapsto$ 0.025 and set counter = 1*
5: *Pick an Observation X_{test} from TestSet*
6: ***DO*** *counter $\mapsto$ counter + 1*
7: *In the pattern unit, compute unconditional probability* ***p(***X_{test}***)*** *and conditional probability* ***p(***$X_{test}|c_i$***)*** *respectively as*

$$p(X_{test}) = \frac{1}{(2\pi)^{M/2}\sigma^M M} \sum_{i=1}^{M} \exp\left[-\frac{(X_{i,k} - X_{test})^T (X_{i,k} - X_{test})}{2\sigma^2}\right] \tag{8.1}$$

$$p(X_{test}|c_i) = \frac{1}{(2\pi)^{M/2}\sigma^M N_i} \sum_{i=1}^{N_i} \exp\left[-\frac{(X_{i,k} - X_{test})^T (X_{i,k} - X_{test})}{2\sigma^2}\right] \tag{8.2}$$

where M = total number of observations in the training set, X_{test} = test dataset, $X_{i,k}$ = k_{th} training vector from plant species of class c_i with i = 1(1)32, σ = PNN spread or smoothing parameter.

8: *Compute posteriori probability of X_{test} as*

$$p(c_i|X_{test}) = \frac{p(X_{test}|c_i)p(c_i)}{p(X_{test})}$$

9: *Compute the average of inputs from pattern units as*

$$f_i = \frac{1}{N} \sum_{i=1}^{N} p(c_i|X_{test}) \tag{8.3}$$

where N_i = Number of training patterns belonging to class c_i.

10: ***UNTIL*** *counter = M*
11: *The classification of each pattern vector is made according to the Baye's Rule:*

$$i = argmax\ \{f_i\} \tag{8.4}$$

12: ***end procedure***

9 Experimental Validation

We validated our experiment using 10-fold cross validation. The steps taken in the 10-fold CV are shown in Figure 10. The feature space (dataset) X is partitioned into k subsets that are roughly of the same size. This partitioning may be written as $X = \bigcup_{i=1}^{k} X_i$ where each of the subset is called a fold. Thus, there are k folds derived from partitioning the original set (feature space) X. The PNN is trained on $k-1$ folds while the k_{th} fold is used for testing. The procedure is repeated such that each subset (fold) is used only once for testing (See Figure 10). The generally recommended value for k is 5 or 10. The k choice for this study is 10. The fascinating merit of the k-Fold CV is that all the observations in the original dataset (feature space) are eventually used for both training and testing. The CV method is much more accurate determinant of the classifier The accuracy of the PNN classifier was computed as

$$PNN_{accuracy} = \frac{trace(ConfuseMatrix)}{sum(ConfuseMatrix)} \tag{9.1}$$

where $trace(.)$ is the sum of all the elements in the backward diagonal, and $sum(.)$ is the sum of all the entries in **ConfuseMatrix**. The confusion matrix is a tabular tool or matrix display of the instances from the training set that were correctly and incorrectly predicted by the (PNN) classifier. It can be represented as **ConfuseMatrix** $\in R^{c \times c}$, a square matrix whose (backward) diagonal elements depicts the actual classification accuracy and c is the number of classes in the dataset.

10 Results

The results of this experiment proved that Zernike moments are more effective than geometric moments. Figures 5,1,6,7 and Table 2 show that ZM remained constant despite rotation, scaling, and translation. Using ZM alone gave accuracy of 91%, while the geometric features gave 89%. The GA-selected features improved the accuracy of the classifier from 91% to 92.1 % since it combined both features together in an efficient manner. The 10-fold CV has also been useful in eradicating biasness in our classification system. The 10-fold CV result for our system was 90.05%.

11 Conclusion

We have demonstrated the effectiveness of Zernike moment in image classification system. A neuro-genetic intelligent system has been built with PNN classifier. The feature extracted viz ZM and Geometric features were further subjected to GA to bring the best features for optimal accuracy. The 10-fold CV used enabled the whole system to be unbiased. A demonstration of affine properties of ZM are comprehensivel stated and explained. Several Figures showing the TRS invariant properties of the ZM are shown. The ZM proved more efficient than the geometric features. The numbers of training sample available for each 32 species of plants used are shown in Table 4 with number of incorrect classifications.

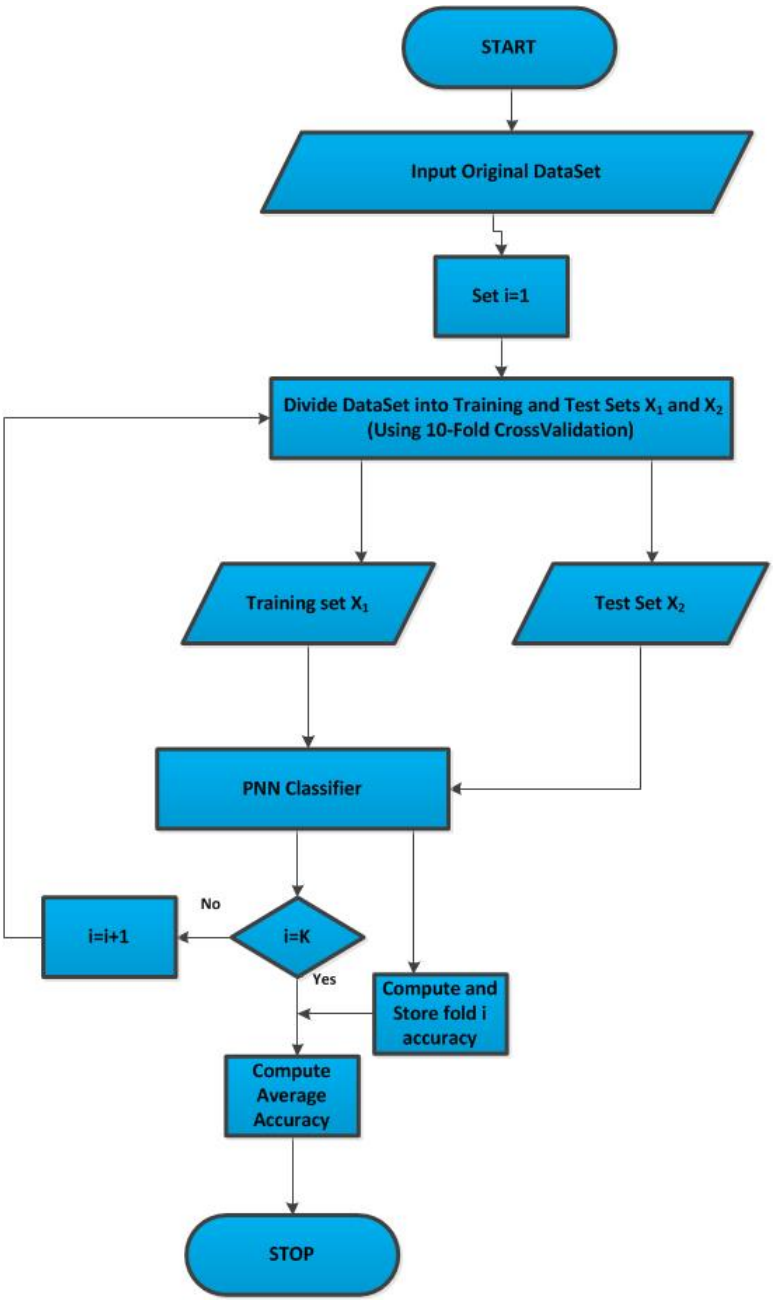

Figure 10: Visual Representation of 10-Fold Cross Validation Experiments. The 10-Fold CV runs for 10 iteration, computing the classification accuracy for each fold , storing the accuracies and finally computing the average of these accuracies.

Table 4: Names of Plant Species in the Flavia Dataset
SN = Scientific Name, TS = Training Samples, WCN = Wrong Classification Number

SN	**TS**	**WCN**
Phyllostachys Edulis	59	0
Aesculus Chinensis	63	1
Berberis Anhweiensis	65	2
Cercis Chinensis	72	2
Indigofera Tinctoria	73	0
Acer Palmatum	56	4
Phoebe Nanmu	62	3
Kalopanax Septemlobus	52	0
Cinnamomum Japonicum	59	0
Koelreuteria Paniculata	55	0
Ilex Macrocarpa Oliv	50	1
Pittosporum Tobira	63	0
Chimonanthus Praecox	52	0
Cinnamomum Camphora	65	0
Viburnum Awabuki	60	2
Osmanthus Fragrans	56	0
Cedrus Deodara	77	0
Ginkgo Biloba	62	1
Lagerstroemia Indica	61	1
Nerium Oleander	66	1
Podocarpus Macrophyllus	60	1
Prunus Serrulata	55	0
Ligustrum Lucidum	55	1
Tonna Sinensis	65	0
Prunus Persicapeach	54	0
Manglietia Fordiana Oliv	52	0
Acer Buergerianum Miq	53	0
Mahonia Bealei	55	2
Magnolia Grandiflora	57	1
Populus Canadensis	64	0
Liriodendron Chinense	53	1
Citrus Reticulata	56	0

Figure 11: Image Classification Using ZM and Geometric Features

Competing Interests

The authors declare that no competing interests exist.

References

[1] Nixon MS, Aguado AS. Feature Extraction and Image Processing for Computer Vision. Elseviet Ltd, the Boulevard, Langford Lane, Kindlington, Oxford, OX5 1GB, UK; 2012.

[2] Flusser J. On the Independence of Rotation Moment Invariants. Pattern Recognition. 2000;33:1405-1410.

[3] Flusser J, Suk T, Zitova B. Moments and Moment Invariants in Pattern Recognition. A John Wiley and Sons, Ltd, Publication. 2009;1-303.

[4] Simon XL. Image Analysis by Moments. PhD thesis at the department of Electrical and Computer Engineering, The University of Manitoba Winnipeg, Manitoba, Canada; 1993.

[5] Teague MR. Image analysis via the general theory of moments. J. Optical Soc. Am. 1980;70:920930.

[6] Chen BJ, Shu HZ, Zhang H, Chen G, Toumoulin C, Dillenseger JL, Luo LM. Quaternion Zernike moments and their invariants for color image analysis and object recognition. Signal Processing. 2012;92(2):308-318.

[7] Thawar A, Zyad S, Lala K, Sami B. Object Classification via Geometric, Zernike and Legendre Moments. Journal of Theoretical and Applied Information Technology. 2009;7(1):31-37.

[8] Hasan SY. Study of Zernike moments using analytical Zernike polynomials. Advances in Applied Science Research. 2012;3(1):583-590

[9] Sabhara RK. Lee Chin-Poo. Lim Kian-Ming. Comparative Study of Hu Moments and Zernike Moments in Object Recognition. Smart Computing Review. 2013;3(3):166-175.

[10] Xin Y, Miroslaw P, Simon XL. Image Reconstruction with Polar Zernike Moments. ICARPR 2005, LNCS 3687. 2005;394-403.

[11] Zhao Y, Wang S, Fend G, Tang Z. A Robust Image Hashing Method Based on Zernike Moments. Journal of Computational Information Systems. 2010;6(3):717-725.

[12] Aliaga Daniel G. Color and Perception. CS635 Spring 2010, Department of Computer Science Purdue University. 2010;1-34.

[13] Wu SG, Bao FS, Xu EY. Wang Yu-Xuan. Chang Yi-Fan. Xiang Qiao-Liang. A Leaf Recognition Algorithm for plant Classification Using Probabilistic Neural Network. IEEE 7th International Symposium on Signal Processing and Information Technology, Cario, Egypt; ArXiv 0707.4289 v1 [CS.AI]; 2007.

[14] Russ JC. The Image Processing Handbook. CRC Press; 2011. Boca Raton

[15] Abdul K. Lukito. E N. Adhi S. Santosa. P I. Experiments of Zernike Moments For Leaf Identification. Journal of Theoretical and Applied Information Technology. 2012;41(1):83-93.

[16] Kadir A. Neural Network Application on Foliage Plant Identification. Indonesia; 2011.

[17] Nixon MS, Aguado AS. Feature Extraction and Image Processing. Newnes; 2002.

[18] Bruzzone L, Persello C. A novel approach to the selection of robust and invariant features for classification of hyperspectral images. Department of Information Engineering and Computer Science, University of Trento; 2010.

[19] Melanie M. An Introduction to Genetic Algorithms. A Bradford Book The MIT Press; 1999.

[20] Tian J, Hu Q, Ma X, Ha M. An Improved KPCA/GA-SVM Classication Model for Plant Leaf Disease Recognition. Journal of Computational Information Systems. 2012;18(8):7737-7745.

[21] Sivanandam SN, Deepa SN. Introduction to Genetic Algorithms. Springer-Verlag, Berlin, Heidelberg; 2008.

[22] Specht D.F. Probabilistic Neural Networks for Classification, Mapping, or Associative Memory. IEEE International Conference on Neural Networks. 1988; 1; 525-532.

[23] Yenugu M. Fisk J.C. Marfurt K.J. Probabilistic Neural Network inversion for characterization of coalbed methane. In 2010 SEG Annual Meeting. Society of Exploration Geophysicists 2010; 1-2.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=550&id=6&aid=4884