2015

A neuronal classification system for plant leaves using genetic image segmentation

Oluleye Babatunde
Edith Cowan University

Leisa Armstrong
Edith Cowan University, l.armstrong@ecu.edu.au

Dean Diepeveen

Jinsong Leng
Edith Cowan University, j.leng@ecu.edu.au

10.9734/BJMCS/2015/14611
Originally published as: Babatunde,O.H., Armstrong, L., Diepeveen, D., & Leng,J. (2015). A neuronal classification system for plant leaves using genetic image segmentation. British Journal of Mathematics & Computer Science. 9(3) 261-278. Original article available here

British Journal of Mathematics & Computer Science
9(3): 261-278, 2015, Article no.BJMCS.2015.201
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

A Neuronal Classification System for Plant Leaves Using Genetic Image Segmentation

Babatunde Oluleye[1,2*], Armstrong Leisa[1], Diepeveen Dean[3] and Leng Jinsong [4]

[1] *School of Computer and Security Science, Edith Cowan University, Perth, Western Australia.*
[2] *Department of Information and Communication Technology, Osun State University, Osogbo, Osun State, Nigeria.*
[3] *Department of Agriculture and Food, Perth, Government of Western Australia.*
[4] *Security Research Institute, Edith Cowan University, Perth, WA, Australia.*

Article Information

DOI: 10.9734/BJMCS/2015/14611
Editor(s):
(1) Victor Carvalho, Polytechnic Institute of Cvado and Ave, Portuguese Catholic University and Lusiada University, Portugal.
(2) Tian-Xiao He, Dept of Mathematics and Computer Science, Illinois Wesleyan University, USA.
Reviewers:
(1) Anonymous, Chongqing University of Technology, China.
(2) Gulshan Kumar, Dept of Computer Applications, SBS State Technical Campus, India.
(3) Anonymous, Malaysia.
(4) Anonymous, Lebanon.
(5) Anonymous, Lodz University of Technology, Poland.
Complete Peer review History:
http://www.sciencedomain.org/review-history.php?iid=1144&id=6&aid=9437

Original Research Article

Received: 08 October 2014
Accepted: 11 May 2015
Published: 26 May 2015

Abstract

This paper demonstrates the use of radial basis networks (RBF), cellular neural networks (CNN) and genetic algorithm (GA) for automatic classification of plant leaves. A genetic neuronal system herein attempted to solve some of the inherent challenges facing current software being employed for plant leaf classification. The image segmentation module in this work was genetically optimized to bring salient features in the images of plants leaves used in this work. The combination of GA-based CNN with RBF in this work proved more efficient than the existing systems that use conventional edge operators such as Canny, LoG, Prewitt, and Sobel operators. The results herein showed that GA-based CNN edge detector outperforms other edge detector in terms of speed and classification accuracy.

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

Keywords: Radial basis networks; cellular neural networks; genetic algorithm.
2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

1 Introduction

Plants are traditionally recognised by manual matching of the plants features such as leaves, flowers, and bark [1]. In view of the large number of plant species available [2, 3], the manual approach to plant classification is slow and prone to human error. There are several techniques which are currently being employed to build computer-based vision systems using features of plants extracted from images as input parameters to various classifier systems [4], [5], [6]. In this paper, a technique to argument already existing techniques of plant leaves identification system is described. The main contribution of this paper is to increase the classification speed and accuracy of the existing systems by incorporating Genetic Cellular Neural Networks for image segmentation.

2 Related Works

The following authors and their associated techniques (in Table 1) have been reported.

Table 1: Some existing and recent works on plant recognition systems

Author(s)	**Techniques**	**Features**
[7]	Image Pre-Processing, Moment Invariants, General Regression Neural Network	Leaf Image Moments
[8]	Aspect Ratio, Rectangularity, Convex Area Ratio, Convex Parameter Ratio, Sphericity, Circularity, Eccentricity, Form Factor, Regional Moments Inertia, Angle Code Histogram	Geometric features
[9]	Probabilistic Neural Network (PNN), Image Pre-Processing, Principal Component Analysis (PCA)	Geometric features
[3]	K-nearest neighbor (k-NN),Image pre-Processing,Aspect ratio, Roundness Ripples features, Ripples Counting,Ripples Pixel Counting, Half-leaf Area Ratio, Upper Leaf Area Ratio, Lower Leaf Area Ratio, Colour Features, Vein Features, Threshold	Geometric features
[10]	Fuzzy Logic Selection, Neural Networks,Image Pre-Processing, Principal Component Analysis	Geometric features
[11]	Fuzzy Segmentation,Image Pre-Processing, Wavelet Transformation	Leaf Image Moments
[12]	Thresholding method, H-Maxima transformation, Moment-Invariants, Centroid-Radii and Neural Networks classifiers	Moment Images
[13]	Image Segmentation, Wavelet Transform, Gaussian Interpolation,K-Nearest Neighbor(K-NN),(1-NN),Radial Basis Probabilistic Neural Networks.	Geometric Features,Leaf Image Moments
[14]	Douglas-Peuker Algorithm(Shape Polygonal Approximation, Invariant Attributes), Genetic Algorithm, kNN	Image moments

[15]	Centroid-Contour Distance (CCD) curve, Eccentricity and Angle code histogram (ACH).	Geometric Features
[16]	Color Metrics, Edge Histogram Computation, Image Pre-Processing	Colour moments
[17]	Basic Grey Level Aura Matrix (BGLAM) technique and Statisitical Properties of pores distribution(SPPD) for wood features.	Colour moments
[18]	Image pre-processing (shadow,background correction, binarization), petiole removal , Ellipse based Blob Ranking, GrabCut leaf segmentation, Random Forest Classifier	Tooth Features and Morphological features as found in ([9])
[19]	Inner Distance Shape Context (IDSC), K-NN, Color image segmentation	Shapes of plants leaves

In the existing works summarized in Table 1, there is need to improve the accuracy of the classifcation model. One way to do this is do employ efficient image segmentation techniques. Once the images are efficiently segmented, the features subsequently generated may help improve the classification accuracy of the system. In this paper, a new classification model involving RBF, genetic algorithm (GA) and cellular neural networks (CNN) was employed to develop a computer-based vision system for automatic identification of plant species.

3 Radial Basis Networks (RBF)

A radial basis function network is a variant of artificial neural network (ANN) that employ radial basis functions as activation functions [20], [21], [22]. RBF networks are extensively being used in research because they are universal approximator with compact topology. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. The input to RBF could be a feature set or a vector of real numbers $\mathbf{x} \in \mathbb{R}^n$. All the elements of the input vector (s) are algebraically mapped to a scalar output as $\phi : \mathbb{R}^n \to \mathbb{R}$ using the functional in Equation 3.1.

$$\phi(\mathbf{x}) = \sum_{i=1}^{K} w_i \rho(||\mathbf{x} - \mathbf{c}_i||) \tag{3.1}$$

where K is the number of neurons in the hidden layer, $\mathbf{c}_i$ is the center vector for neuron i, w_i is the weight of neuron i in the linear output neuron and ρ is a radius-based metric. A radial basis function is a real-valued function whose value depends only on the distance from the origin. Thus $\phi(X) = \phi(||X||)$. Functions that depend only on the distance from a center vector are radially symmetric about that vector, hence the name radial basis function (see appendices A1, A2 and A3). In the basic form all inputs are connected to each hidden neuron. An RBF can be trained without back propagation since it has a closed-form solution. The neurons in the hidden layer contain basis functions. A common basis function for RBF network is a kind of Gaussian function without scaling factor. More detailed about RBF can be found in [23], [24], [25], [20], [26], [27].

4 Cellular Neural Networks

Cellular Neural Networks (CNN) are variants of Artificial Neural Networks (ANN), which are also dynamical systems in which all the elements are locally connected through a neighborhood

communication [28]. The dynamical model (see Equation 4.1) representing CNN was modelled from an electrical circuit (see figure 1) in 1988 by Chua and his graduate student [29], [30]. A standard CNN topological structure is made up of an $M \times N$ or rectangular array of cells $C(i,j)$(or dynamic components) with Cartesian coordinates (i,j), $i = 1(1)M, j = 1(1)N$ and $M, N \in \mathbb{Z}^+$. CNN is a hybrid model, since its features share from both Cellular Automata and Artificial Neural Networks [31], [32]. The circuit structure and element values of all cells of a CNN are homogenous.

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{(k,l)\in\mathcal{N}(i,j)} Ay + \sum_{(k,l)\in\mathcal{N}(i,j)} Bu + I \quad (4.1)$$

where $Ay = A(i,j;k,l) \cdot y_{kl}(t), Bu = B(i,j;k,l) \cdot u_{kl}$ and $I = I(i,j;k,l)$

$$y_{ij}(t) = f(x_{ij}(t)) = \frac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \quad (4.2)$$

The variables appearing in Equations 4.1 and 4.2 are defined in the reference papers [28] and [32].

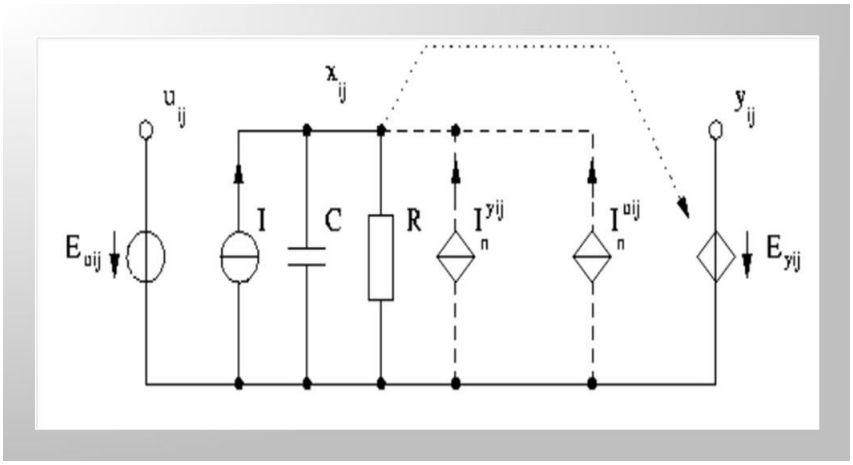

Figure 1: Circuit representing cellular neural networks

5 Experimental Methodology

The steps in the experimental research are provided in figure 7.

5.1 The Flavia Dataset

The images of leaves of plant species used in this study are found in the Flavia dataset which is publicly available [9]. The Flavia dataset is a constrained set of leaf images taken against a white background and without any stem present. The leaves in the dataset have a varying number of instances as shown in [33]. The dataset has 1907 images of 32 species of plants. For this study, the dataset was divided into two disjoint sets, each of which contains 1587 images and 320 images for both training and test set respectively. The shapes of one sample for each species of plant in the Flavia dataset are shown in Figure 9.

5.2 Image Pre-processing

The first step (after image acquisition), is to pre-process the images found in [9]. The original images (which are colored i.e rgb images), are first converted to grayscale images using the formular in Equation 5.1. The R, G, and B in the equation respectively represents the red, green, and blue components of the colored image [33], [31].

$$RGB2GRAY = 0.299R + 0.587G + 0.114B \tag{5.1}$$

5.3 Image Segmentation Using Genetic CNN

The goal of image segmentation is to find regions of interest (ROI) in the images. The division of images into ROIs is often necessary before any processing can be done at a higher level than that of the pixel [34]. Thus, image segmentation is the division of an image into disjoint regions such that the intersection of differently indexed regions is null and the union of all regions is the image itself. Three main classes of image segmentation are (i) region growing, (ii) clustering methods, and (iii) boundary detection. The CNN herein falls into the third category. Next to image pre-processing, a GA (see Table 2) was employed to optimize the CNN edge detection templates in Equation 5.4. Detailed information on GA can be found in [35], [36], [37], [38], [39]. These templates are the matrix coefficient for the systems of equations in 4.1 & 4.2. Finite element methods (FEM) was used in discretizing the ODE in Equation 4.1. The template design process for the CNN is shown in figure 3 using the GA parameters in Table 2. The fitness function used by the GA-CNN process is given in equation 5.2 where I_{ij} and O_{ij} are evaluated and input image pixels respectively. The best chromosome was obtained at generation 101 as shown in figure 6.

$$f(CNN) = 1 - \left[\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (I_{ij} - O_{ij})^2}{8NM} \right] \tag{5.2}$$

The main idea behind the CNN is to input an image and discretize the ODE using an appropriate numerical methods. FEM was used to solve the ODE during both GA simulation and CNN image segmentation. The final templates shown equation 5.4 are then used for image segmentation. As documented by [40] & [41], the relationship between the input image and the segmented image in the CNN is expressed as equation 5.3.

$$dx = -x + conv(x, A) + conv(x, B) + I \tag{5.3}$$

where dx is the edge/segmented image (or ROI), x is the original image. The function $conv$ is a convolution operator. The convolution process requires overlaying the templates masks on the input image, multiplying the coincident values and summing all the results. This is pretty much similar to finding the vector inner product of the mask with the underlying subimage. The elements (or matrices) of the triple $\{A(i,j;k,l), B(i,j;k,l), I_{ij}\}$ are the cloning templates for the CNN Model or ODE in equation 4.1. These three (3) matrices determine the behavior of the CNN [42]. The templates specify how an image $\{f(x_i, y_j) : 1 \leq i \leq M, 1 \leq j \leq N\}$ at time $t = 0$ will be transformed to produce an $M \times N$ output image $y(t)$ for $t \geq 0$. The outputs of the segmentation stage are finally passed on to feature extraction modules has functions for Fourier Descriptors (FD) and Zernike Moments (ZM).

CNN Templates:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}, I = (-1) \tag{5.4}$$

CNN Templates (Genetically optimized):

$$\mathbf{A} = \begin{pmatrix} -0.0188 & -7.2196 & -1.6024 \\ -2.2306 & 20.8999 & -2.2306 \\ -1.6024 & -7.2196 & -0.0188 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} -0.0397 & 0.3402 & -0.0362 \\ -0.2233 & -0.2497 & -0.2233 \\ -0.0362 & 0.3402 & -0.0397 \end{pmatrix}, I = (-3.3014) \tag{5.5}$$

Sobel Operator:

$$\mathbf{H_x} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \mathbf{H_y} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{5.6}$$

Prewitt operator:

$$\mathbf{H_x} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{H_y} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \tag{5.7}$$

Laplacian operator:

$$\mathbf{L_1} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{L_2} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & - & -1 \end{pmatrix} \tag{5.8}$$

The time required by the the conventional edge operatures were longer than the time required by the GA-based CNN templates. It's to be noted that GA was not involved in the the image segmentation process, rather it was used separately to obtain optimal edge templates shown in Equations 5.5. Figure 2 shows the MATLAB-based GUI developed to extract optimal edge from the images of the plants in the Flavia dataset.

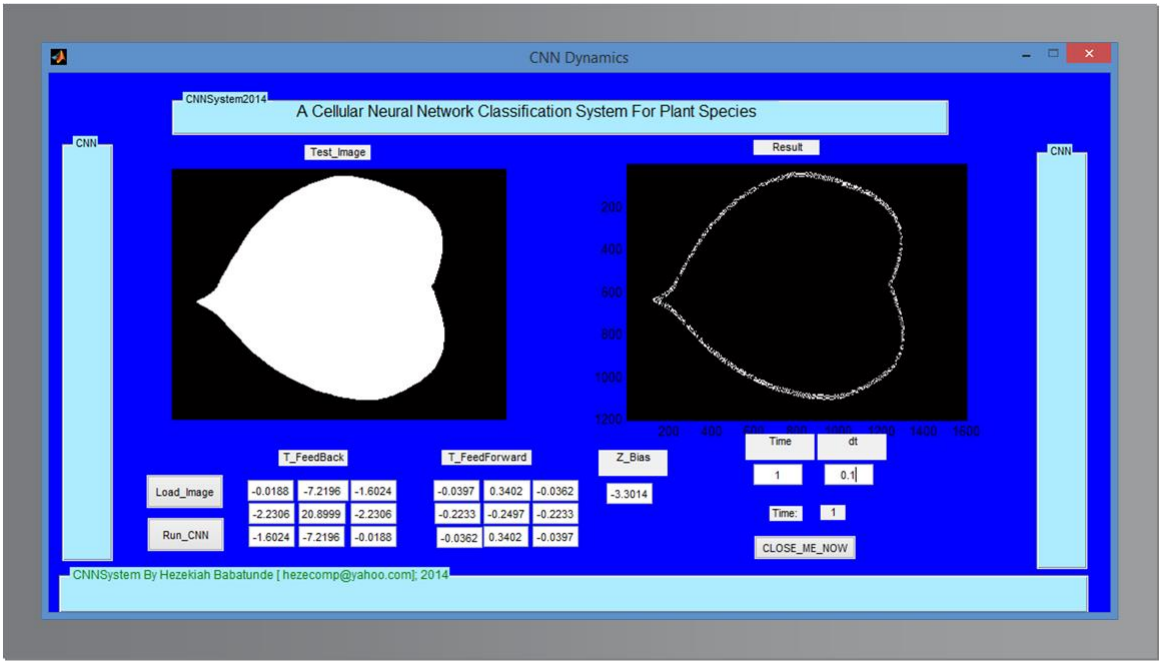

Figure 2: Genetic CNN system for edge detection

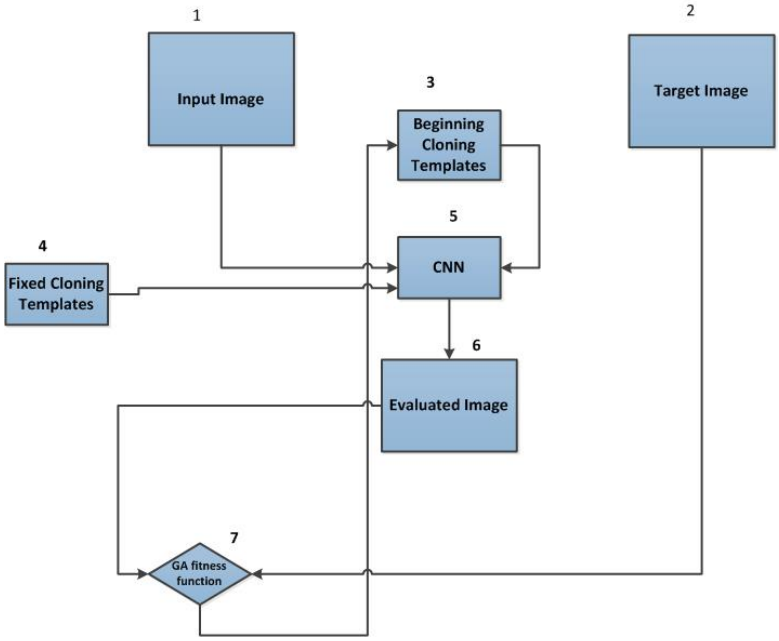

Figure 3: Genetic Cellular Neural Networks

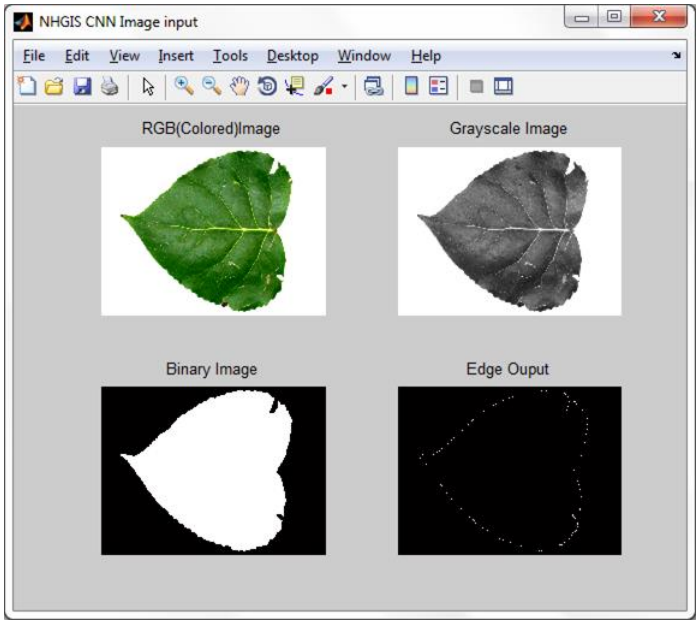

Figure 4: Image pre-processing and segmentation

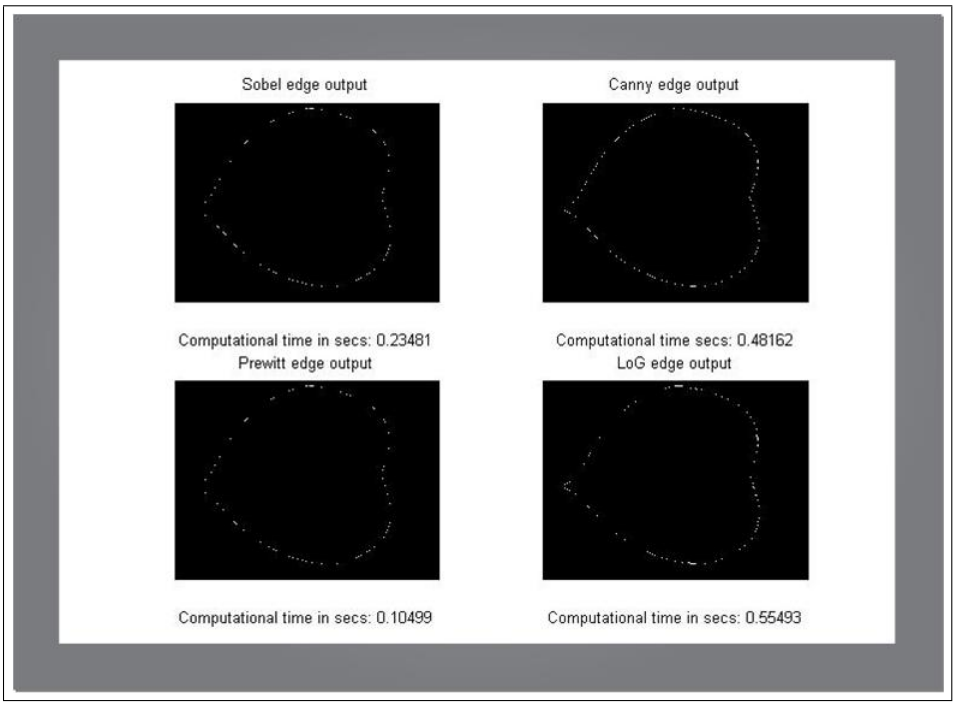

Figure 5: Edge outputs from conventional edge operators

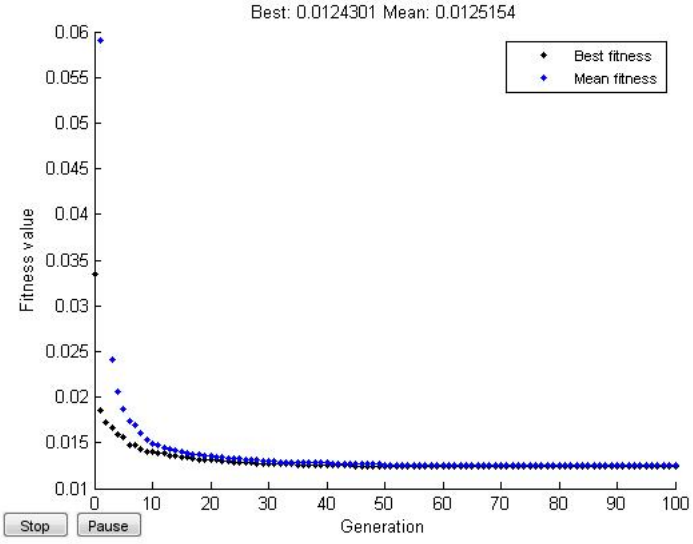

Figure 6: Fitness plot for the GA

Table 2: Configuration for the GA

GA Parameter	**Value**
Population size	50
Genomelength	19
Population type	real
Fitness Function	function f(.) in equation 5.2
No of generations	100
No of GA Iteration	1
Crossover	Heuristic Crossover
Crossover Fraction	0.8
Mutation	Uniform Mutation
Mutation Fraction	0.01
Selection scheme	Tournament of size 2
EliteCount	2

5.4 Features Extraction

The outputs of the genetic segmentation process are used as parameters for Fourier Descriptors (FD) and while binary version of the images are used by Zernike Moments (ZM). The detailed about ZM and FD can be found in [33] and [43] respectively.Twenty features were extracted from the Flavia dataset disussed in section 5.1 . These features are represented in Table 3.

Table 3: 20 features extracted from the Flavia Dataset

Observation	**F1 F2 F3 F4 F5 F6 F7 F8 F9 ...F20**
Image1	$X_{1,1}$ $X_{1,2}$ $X_{1,3}$ $X_{1,4}$ $X_{1,5}$... $X_{1,20}$
Image2	$X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{2,4}$ $X_{2,5}$... $X_{2,20}$
Image3	$X_{3,1}$ $X_{3,2}$ $X_{3,3}$ $X_{3,4}$ $X_{3,5}$... $X_{3,20}$
...	...
...	...
...	...
Image1907	$X_{1907,1}$ $X_{1907,2}$ $X_{1907,3}$... $X_{1907,20}$

5.5 Learning System Based on Radial Basis Networks

The steps involved in the design of the classification system shown in Figure 8 are described in Figure 7. These steps are similar to most image classification systems. The distinguishing features of the model lies in the application of genetic CNN for image segmentation. Each node in the input layer of the RBF corresponds to a feature vector from Table 3. The second layer is the only hidden layer in the RBF network. The second layer applies non-linear mapping from input vector space into hidden layer space through appropriate non-linear function such as guassian kernel (see equation 5.9) which was used in this work. The x in equation 5.9 is the training sample, c_i is the hidden ith neuron and σ is the width of the basis function which is a multiple of the average distance between the centers in the RBF network. The σ determines the receptive width of the RBF. The output layer is made up of neurons that are directly connected to the hidden layer neurons [44]. The output value for the training set or any input to the RBF is expressed as equation 5.10. The w in equation 5.10 is the weight factor normally computed as $w = (h^T h)^{-1} C$ where C is the target class matrix and h^T means "Transpose of h". The number of neurons in the output layer is the same as the number of classes in the dataset while the number of neurons in the input layer is

equal to the number of features in the training set. For this work, the number of features is 20. GA was used in determining the σ value while K-means clustering was used in forming the centers in the hidden layer. The entire implementation (including the GUI) was done using MATLAB 2013a.

$$h(||x - c_i||) = \exp\left(-\frac{||x - c_i||^2}{2\sigma^2}\right) \tag{5.9}$$

$$y(x) = \sum_{i=1}^{K} w_i * h(||x - c_i||) + b \tag{5.10}$$

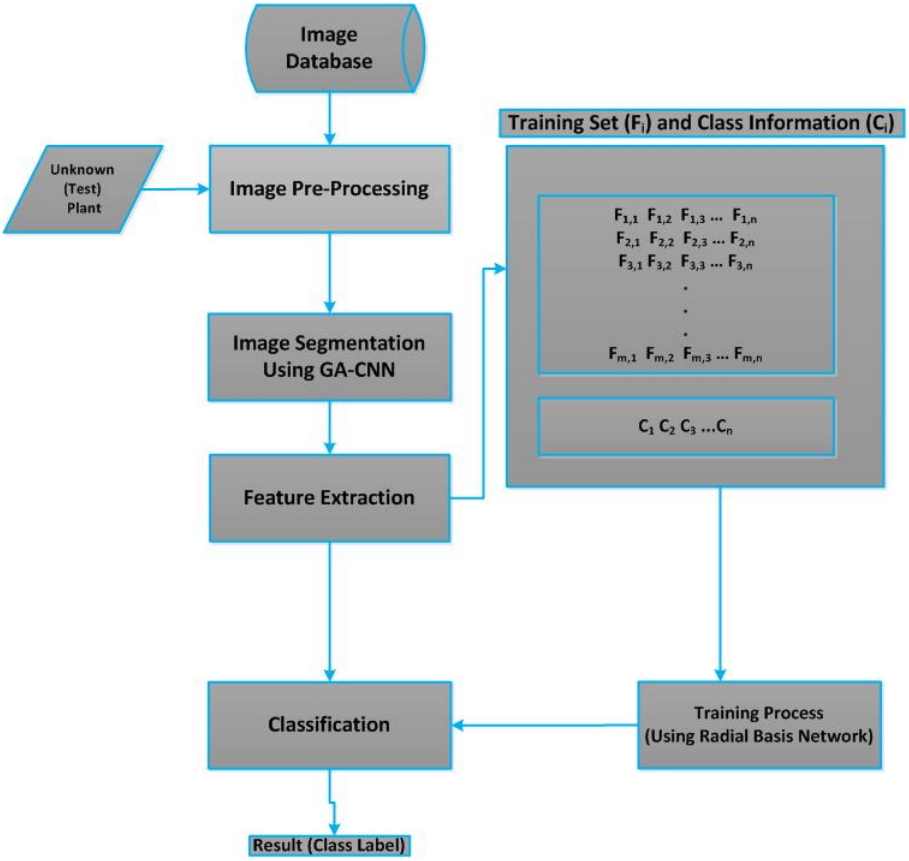

Figure 7: Learning system based on Radial Basis Network

6 Experimental Validation

The approach used in validating the RBF herein is the k-Fold Cross Validation (k-Fold CV) with k = 10. Generally, a cross validation (CV) is a method of partitioning the feature space into training and testing sets as shown in [33]. Herein, the naive RBF was fitted using training set, while the fitted model was validated through testing set by measuring the error predicted. The training set and testing set were both disjoint to ensure that the testing set for evaluating the RBF are not

used in fitting the model. Without the loss of any generality, and suppose the information in Table 3 are true, the dataset (feature space) X is then partitioned into two sets viz $X = X_1 \cup X_2$, such that k elements are in X_1 and $D - k$ elements in X_2. The RBF was then trained or fitted using the set X_2. The historical pattern of X_2 was used to produce classifications (predictions) results for observations $X_1{}^* \in X_1$ given X_2. The algorithm for the 10-fold CV is given in Algorithm 6.1 below:

Algorithm 6.1. *k-fold Cross Validation for RBF Classifier*

1: ***procedure*** kFoldRBF(DataSet, k)
2: ***Input*** *DataSet and the class information.*
3: *Partition DataSet into K folds (disjoint sets) using the class information such as* $X = X_1 + X_2$, *where* $(X, c_i) \in \mathbb{R}^D \times \{1, 2, 3, ..., 32\}$
4: ***DO*** *counter* $\mapsto$ *counter + 1*
5: *Remove k and train RBFClassify using feature from all classes except class k*
6: *Use* X_2 *for validation and* X_1 *for Training*
7: *Compute ErrorRBF on the validation set* X_2 *as*

$$Error_{RBF}(X) = RBFClassify(X_2)$$

j = *number of datapoints in the partition k*
8: ***UNTIL*** *counter* = K

$$CV\,Error = \frac{1}{N}\sum_{j=1}^{N} ErrorRBF$$

9: ***end procedure***

7 Results and Discussion

The results of this work are shown in Table 4. Several image segmentation techniques including canny, prewitt, LoG were compared with the genetic CNN used in this work. The matrices describing other edge operators are shown in equations (5.6, 5.7, 5.8). In the diagram shown in figure 4, the edge outputs were the only region of interest (ROI) sent to FD feature extraction module of the system. The ZM accepted only monochrome version of the sample images. The modus operandi of the GA used to optimize both RBF and GA are shown in figure 10. The average width of the RBF neurons is 0.45 with best accuracy occuring at $\sigma = 0.9$. The GA enabled the CNN to bring out salient features from the images used. The CNN was solved using finite element method (FEM). FEM have been known to reduce discretization errors from dynamic systems more often than runge-kutta method. The whole system was unbiasedly evaluated using CV with k = 10 (see Algorithm 6.1). Accuracy and computational time (in seconds) were used as performance metric for the developed system. It's shown in the table that the system performed better than the conventional image segmentation techniques (edge detectors). The CNN may also serve other purposes such as solution of partial differential equation (PDE), ordinary differential equation (ODE) and maximum likelihood estimation in signal processing. When integrated into the classification system, the CNN was found to improve both accuracy and speed of the developed system shown in Table 4.

Table 4: Classification accuracy and computational time

Edge Detector	Time (seconds)	Accuracy (%)
CNN	7.77	90.45
Sobel	8.45	89.32
Canny	8.18	90.05
LoG	8.89	88.45
prewitt	8.76	88.98

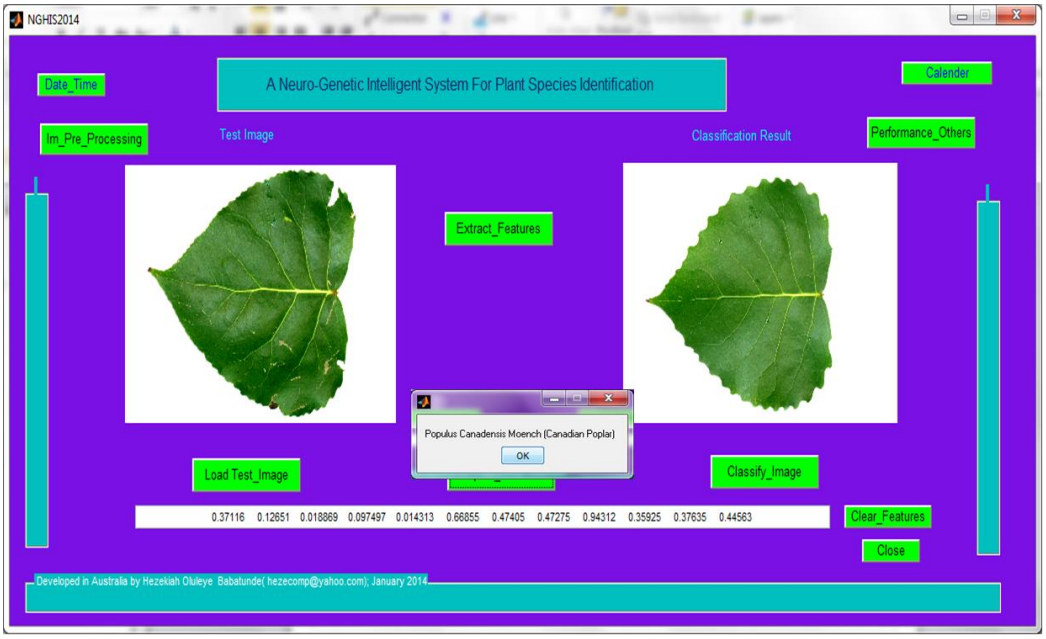

Figure 8: Image classification system based RBF and Genetic segmentation (adapted from [31])

8 Conclusion and Future Works

A demonstration of novel neuro genetic intelligent system has been done in the work described in this paper. The system was found to perform better when compared to conventional image segmentation techniques (edge detectors) as evidenced by reduction in computational time and increased classification accuracy (see Table 4). This work is adaptive and may further be extended by using optimization techniques apart from GA, to determine it weights and centres and make some analysis and comparison.

Acknowledgements

This work was jointly supported by Edith Cowan University International Postgraduate Research Scholarship (ECUIPRS), Australia and Tertiary Educational Trust Funds (TETF) of Nigeria.

Competing Interests

The authors declare that no competing interests exist.

References

[1] Meeta K, Mrunali K, Shubhada P, Prajakta P, Neha B. Survey on techniques for plant leaf classification. International Journal of Modern Engineering Research (IJMER), 2012;1(2):538-544.

[2] Pornpanomchai Chomtip, Kuakiatngam Chawin, Supapathranon Pitchayuk, Siriwisesokul Nititat. Leaf and flower recognition system (e-botanist). IACSIT International Journal of Engineering and Technology. 2011;3(4):10-15.

[3] Chomtip P, Supolgaj R, Piyawan T, Chutpong C. Thai herb leaf image recognition system (THLIRS). Kasetsart J. (Nat. Sci.). 2011;45:551-562.

[4] Babatunde O, Armstrong L, Leng J, Diepeveen D. A survey of computer-based vision systems for automatic identification of plant species. Journal of Agricultural Informatics. 2015;6(1):61-71.

[5] James S Cope, David Corney, Jonathan Y Clark, Paul W Remagnino. Plant species identification using digital morphometrics: A review. Digital Imaging Research Centre, Kingston University, London, UK and Department of Computing, University of Surrey, Guildford Surrey, UK. 2011;1-21.

[6] Pahalawatta KK. A plant identification system using both global and local features of plant leaves. MSc Thesis at the department of Computer Science and Software Engineering, University of Canterbury, New Zealand. 2008;1-127.

[7] Zalikha Z, Puteh S, Itaza S, Mohtar A. Plant identification using moment invariants and general regression neural network. 11th International Conference on Hybrid Intelligent Systems (HIS). 2011;430-435.

[8] David K, James P, Mathew P. Authomatic plant leaf classification for a mobile field guide (an android application). Department of Electrical Engineering, Stanford California; 2012.

[9] Wu Stephen Gang, Bao Forrest Sheng, Xu Eric You, 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.

[10] Panagiotis T. Plant leaves classification based on morphological features and a fuzzy surface selection techniques. Fifth International Conference on Pattern Recognition, Greece. 2005;365-370.

[11] . Valliammal V, Geethalakshmi SN. A novel approach for plant leaf image segmentation using fuzzy clustering; 2011.

[12] Jyotismita C, Ranjan P. Plant leaf recognition using shape based features and neural network classifiers. International Journal of Advanced Computer Science and Applications (IJACSA). 2011;41-47.

[13] Xiao G Ji-Xiang, Xiao-Feng W. Leaf recognition based on the combination of wavelet transform and gaussian interpolation. China; 2005.

[14] Ji-Xiang Du,Xiao-Feng Wang, Xiao Gu. Shape matching and recognition base on genetic algorithm and application to plant species identification. In Advances in Intelligent Computing , Springer Berlin Heidelberg. 2005;282-290.

[15] Wang Z, Chi Z, Feng D. Shape based leaf image retrieval. IEEE Proceedings, (online 20030160); 2003.

[16] Sandeep Arya, Parveen Lehana. Development of a seed analyzer using the techniques of computer vision. International Journal of Distributed and Parallel Systems (IJDPS). 2012;3(1):149-155.

[17] Marzuki K, Yusof R, Anis S, Mohd K. Improved tropical wood species recognition system based on multi-feature extractor and classifier. International Conference on Electrical and Computer Engineering (ICECE), World Academy of Science, Engineering and Technology. 2011;1702-1708.

[18] Arora Akhil, Gupta Ankit, Bagmar Nitesh, Mishra Shashwat, Arnab Bhattacharya. A plant identication system using shape and morphological features on segmented leaves:team iitk, clef 2012. Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, India and Department of Computer Science and Engineering,University of Florida, Gainesville, USA. 2012;1-14,.

[19] Belhumeur PN, Daozheng C, Steven F, David W, Jacobs W, John K, Haibin L, Ida L, Ravi R, Sameer S, Sean W, Lin Z. Searching the world's herbaria: A system for visual identification of plant species. In Computer VisionECCV 2008, Springer Berlin Heidelberg. 2008;116-129.

[20] Broomhead DS, Lowe D. Radial basis functions, multi-variable functional interpolation and adaptive networks. Royal Signals and Establishment Malvern (United Kingdom), No. RSREMEMO- 41-48; 1988.

[21] Haykin S. Redes neurais - princpios e prticas. Bookman; 2001.

[22] Mark J, Orr L. Introduction to radial basis function networks. Recent Advances in Radial Basis Function Networks. Centre for Cognitive Science, University of Edinburgh, Scotland. 1996;1-67.

[23] Ackley DH. A learning algorithm for boltzmann machines. Cognitive Science. 19859;147-169.

[24] Aleksander I. Neural computing a architecture: The design of brain-like machines. North Oxford, London; 1989.

[25] Amari SI. Mathematical foundations of neurocomputing. Proc. IEEE. 1990;78:1443-1463.

[26] Changbing L, Wei H. Application of genetic algorithm-rbf neural network in water environment risk prediction. In Computer Engineering and Technology (ICCET). 2nd International Conference on. 2010;7:239-242.

[27] Kurban T, Besdok E. A comparison of rbf neural network training algorithms for inertial sensor based terrain classification. Sensors. 2009;6312-6329.

[28] Chua LO, Yang L. Cellular neural networks: Theory and applications. IEEE Trans on Circuits and System. 1988;35(10):1257-1272.

[29] Babatunde Oluleye, Armstrong Leisa, Leng Jinsong, Diepeveen Dean. Application of cellular neural networks and naivebayes classifier in agriculture. AFITA 2014, 9th Conference of the Asian Federation for Information Technology in Agriculture, Australia, Perth, 6 - 9 October; 2014.

[30] Chua LO, Roska T. Cellular neural networks and vision computing. Cambridge University Press; 2002.

[31] Babatunde Oluleye, Armstrong Leisa, Leng Jinsong, Diepeveen Dean. On the application of genetic probabilistic neural networks and cellular neural networks in precision agriculture. Asian Journal of Computer and Information Systems. 2014;2(4):90-100.

[32] Hezekiah B, Akinwale AT, Folorunso O. A cellular neural networks- based model for edge detection. Journal of Information and Computing Science. 2010;5(1):003-010.

[33] Babatunde Oluleye, Armstrong Leisa, Leng Jinsong, Diepeveen Dean. Zernike moments and genetic algorithm: Tutorial and application. British Journal of Mathematics and Computer Science. 2014;4(15):2217-2236.

[34] Umbaugh Scott E. Digital image processing and analysis. Human and Computer Vision Applications with CVIP tools. CRC Press: Taylor & Francis Group, Flourida, USA; 2011.

[35] Aarts EHL, Korst JHM. Simulated annealing and boltzmann machines. Wiley, New York; 1989.

[36] Awad M, Chehdi K. Satellite image segmentation using variable hybrid Genetic Algorithm. Wiley International Journal of Imaging Systems and Technology. 2009;19:199-207.

[37] Babatunde Oluleye, Armstrong Leisa, Leng Jinsong, Diepeveen Dean. A genetic algorithm-based feature selection. International Journal of Electronics Communication and Computer Engineering. 2014;5(4), 889-905.

[38] Holland John H, Genetic algorithms. Scientific American; 1962.

[39] Man K, Tank K, Kwong S. Genetic algorithm: Concepts and applications. IEEE Transactions on Industrial Electronics. 1996;43(5):519-534.

[40] Alireza Fasih, Jean Chamberlian Chedjou, Kyandoghere Kyamakya. Cellular neural network trainer and template optimization for advanced robot locomotion, based on genetic algorithm. Transportation Informatics Group, University of Klagenfurt-Austria. 2011;1-6.

[41] Duraisamy M, Duraisamy S. CNN-based approach for segmentation of brain and lung MRI images. European Journal of Scientific Research. 2012;81(3);298-313.

[42] Biey M, Checco P, Gilli M. Bifurcations and chaos in cellular neural networks. Journal of Circuits, Systems and Computers. 2003;12(04):417-433.

[43] Fourier JB. The analytical theory of heat. The University Press; 1878.

[44] Demuth H, Beale M, Hagan M. Neural network toolbox users guide. The MathWorks. Inc.,Natick, MA; 2013.

[45] Babatunde O, Armstrong L, Leng J, Diepeveen D. Comparative analysis of genetic algorithm and particle swam optimization: An application in precision agriculture. Asian Journal of Computer and Information Systems. 2015;3(1):1-12.

APPENDICES

A1

RBFs are typically used in building function approximation of the form $f : x \mapsto \sum_{i=1}^{K} w_i \phi(||x - x_i||)$, where the approximating function $f(.)$ is represented as a sum of K radial basis functions with their associated center x_i and weight w_i.

A2

Typical examples of basis used RBF (taking $d = ||x - x_i||$ & $0 \leq \epsilon \leq 1$) are:

1. Gaussian: $\phi(d) = e^{-(\epsilon d)^2}$
2. Multiquadratic: $\phi(d) = \sqrt{1 + (\epsilon d)^2}$
3. Inverse quadratic: $\phi(d) = \frac{1}{1+(\epsilon d)^2}$
4. Inverse multiquadratic: $\phi(d) = \frac{1}{\sqrt{1+(\epsilon d)^2}}$
5. Polyharmonic spline:
   (a) $\phi(d) = d^n, n = 1, 3, 5, ...$
   (b) $\phi(d) = d^n \ln(n), n = 2, 4, 6, ...$
6. Thin plate spline: $\phi(d) = d^2 \ln(d)$

A3

The distance d in appendices A and C can be generally defined as $d(x - x_i) = \left(\sum_{i=1}^{n} |x - x_i|^n \right)^{1/n}$

A4

RBF networks were first applied in the solution of multivariate interpolation problems and in numerical analysis. RBF can be applied in regression, classification and time series prediction. K-means clustering, RBF, probabilistic neural networks (PNN) and general regression neural networks (GRNN) are all very similar in operations and architecture.

A5

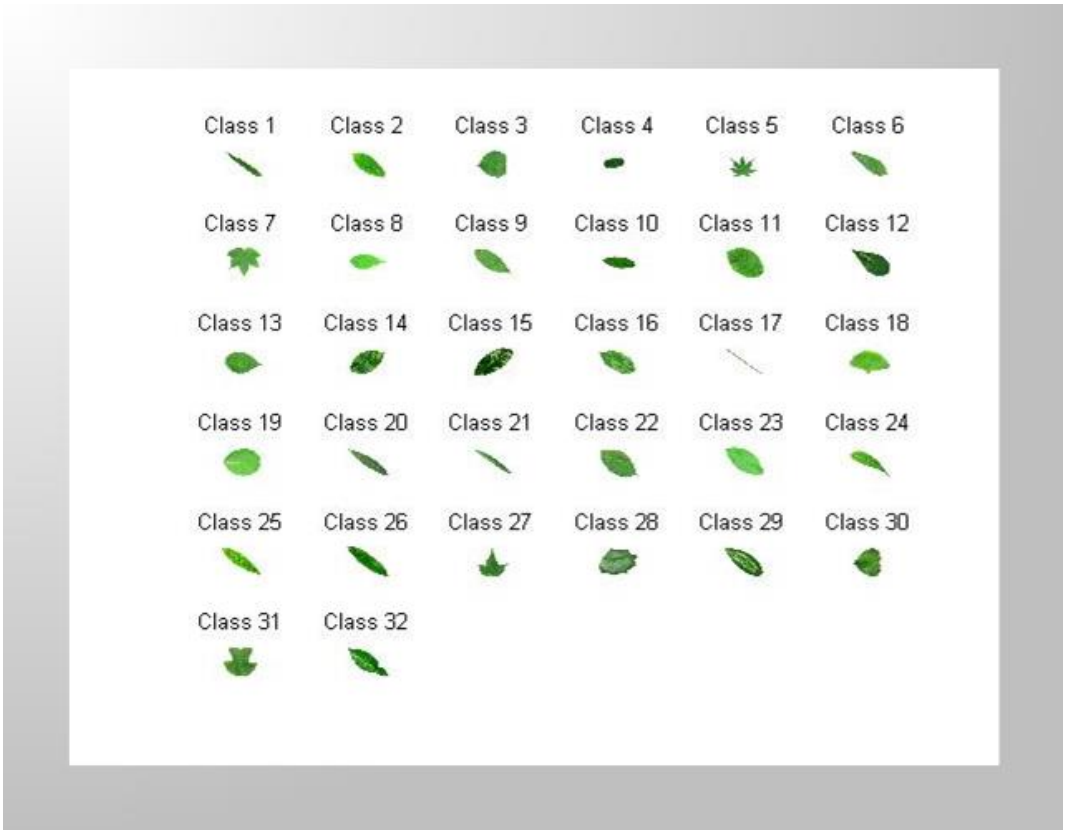

Figure 9: Classes of leaves in the flavia dataset

A6

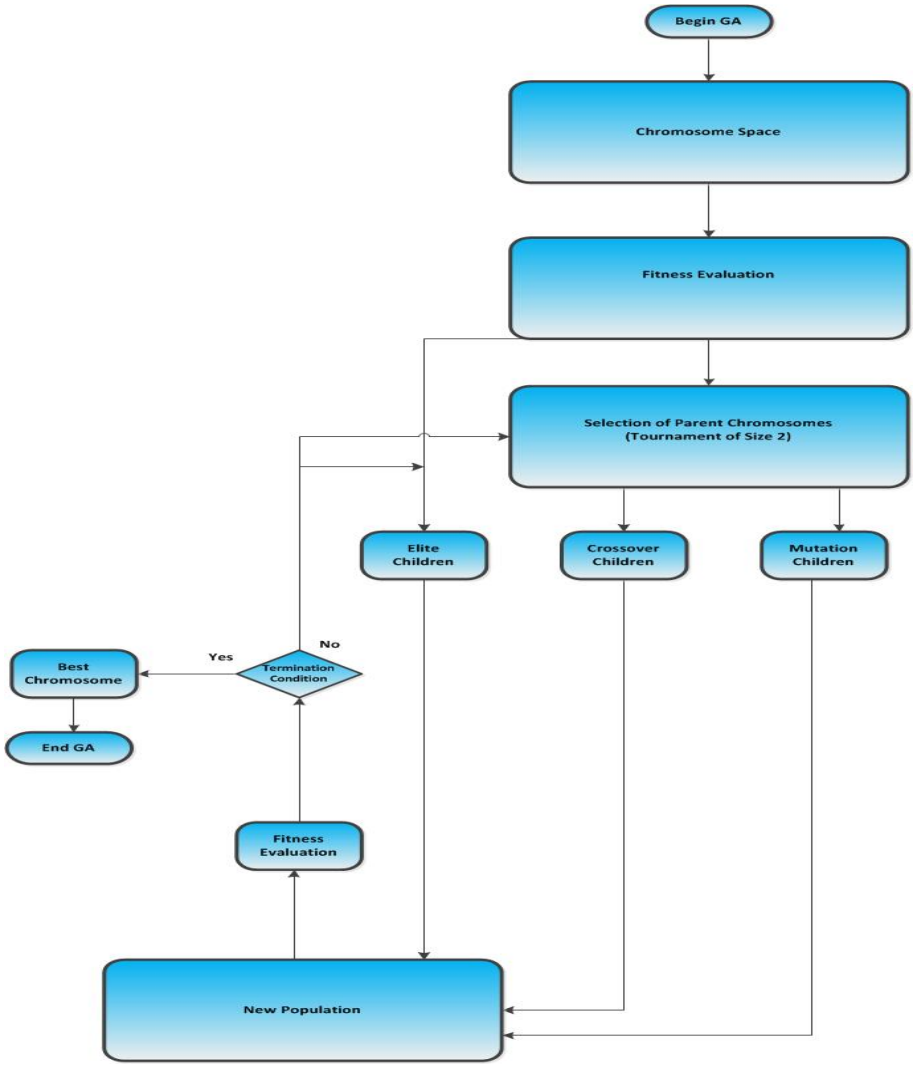

Figure 10: Steps involved in the GA [45]

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=1144&id=6&aid=9437