
A Neuronal Classification System for Plant Leaves using Genetic Image Segmentation

**Original Research
Article**

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.

Keywords: Radial Basis Networks, Cellular Neural Networks, Genetic Algorithm

2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

1 Introduction

Traditional recognition of plants is carried out by manual matching of the plants features such as leaves, flowers, and bark [26]. In view of the large number of plant species available [11], the manual approach to plant classification is slow and prone to human error. As a result of this, attempts to automate this process have been made using features of plants extracted from images as input parameters to various classifier systems [14] & [27]. 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 have been reported. [33, 21, 15, 31, 5, 32, 30, 28, 25]

1. **Jyotismita (et al, 2011)**: Moment invariants, centroid-radii, neural network.

2. **Zalikha (et al, 2011)** : Image pre-processing, moment invariants, general regression neural network
3. **David (et al, 2012)**: Aspect ratio, rectangularity, convex area ratio, convex parameter ratio, sphericity, circularity, eccentricity, form factor, regional moments inertia, angle code histogram & geometric features.
4. **Wu (et al, 2007)**: Probabilistic neural network (PNN), image pre-processing, principal component analysis (PCA) & geometric features.
5. **Arora (et al, 2012)**: Image pre-processing (shadow, background correction, binarization), petiole removal , ellipse based blob ranking, grabcut leaf segmentation, random forest classifier & tooth and morphological features.
6. **Belhumeur (et al, 2008)**: Inner distance shape context (IDSC), K-NN, color image segmentation & shapes of plants leaves.
7. **Xiao (et al, 2005)**: Image segmentation, wavelet transform, gaussian Interpolation, K-Nearest Neighbor, Radial Basis Probabilistic Neural Networks with geometric features and leaf image moments
8. **Wang (et al, 2003)**: Centroid-contour distance (CCD) curve, eccentricity and angle code histogram (ACH) & geometric features
9. **Sandeep (et al, 2012)**: Color metrics, edge histogram computation, image pre-processing & colour moments
10. **Yi-Tou (et al, 2009)**: Rotational invariant methods, grey level co-occurrence matrices (GLCM) & image features, moment images
11. **Marzuki (et al, 2011)**: Basic grey level aura matrix (BGLAM) technique and statisitcal properties of pores distribution (SPPD) for wood features & colour moments.

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 [9, 24, 18]. 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 $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ using the functional in Equation 3.1.

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

where K is the number of neurons in the hidden layer, $\mathbf{c}_i$ is the center vector for neuron i and w_i is the weight of neuron i in the linear output neuron. 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 A, B and C). 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 [9, 10, 22, 2, 3, 4]

4 Cellular Neural Networks

Cellular Neural Networks (CNN) are variants of Artificial Neural Networks (ANN) having neighborhood communication as the main distinguishing feature[13]. CNN was modelled from an electrical circuit (see figure 1) in 1988 by Chua and his graduate student Yang at the department of electrical engineering and computer Sciences, university of California, Berkeley [12]. 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) , $i = 1(1)M, j = 1(1)N$. CNN is a hybrid model, sharing features from both Cellular Automata and Artificial Neural Networks [19, 6]. The circuit structure and element values of all cells of a CNN are homogenous. Equation 4.1 governing the behavior of a CNN cell circuit is a dynamical system (or Ordinary Differential Equation (ODE)) derived from evolution laws and circuit theory.

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

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 [19] and [13].

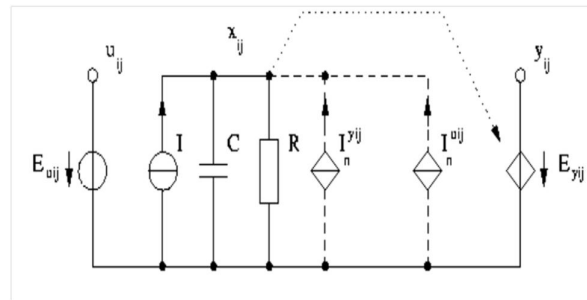


Figure 1: Circuit representing Cellular Neural Networks

5 Experimental Methodology

The steps in the experimental research are provided in figure 5.

5.1 The Flavia Dataset

The source of images of leaves used in this study are found in the Flavia dataset which is publicly available [31]. 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 [7]. 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 32 classes of leaves for plant species used are shown in figure 7.

5.2 Image Pre-Processing

The images found in [31] are first pro-processed. The original colored images are 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.

$$RGB2GRAY = 0.299R + 0.587G + 0.114B \quad (5.1)$$

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 [29]. 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 1) was employed to optimize the CNN edge detection templates in Equation 5.4. Detailed information on GA can be found in [23], [20] and [1]. 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 2 using the GA parameters in Table 1. 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 4.

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

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 Duraisamy2012 and Alireza2011, the relationship between the input image and the segmented image in the CNN is expressed as equation 5.3.

$$dx = -x + conv(x, A) + conv(x, B) + I \quad (5.3)$$

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 [8]. 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:

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

Sobel Operator:

$$\mathbf{H}_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \mathbf{H}_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (5.5)$$

Prewitt operator:

$$\mathbf{H}_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{H}_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad (5.6)$$

Laplacian operator:

$$\mathbf{L}_1 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{L}_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & - & -1 \end{pmatrix} \quad (5.7)$$

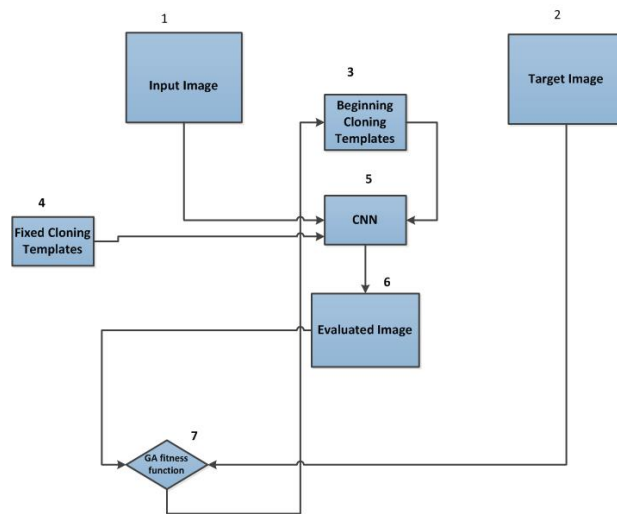


Figure 2: Genetic Cellular Neural Networks

Table 1: Configuration for the GA

GA Parameter	Value
Population size	50
Genomelength	9
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

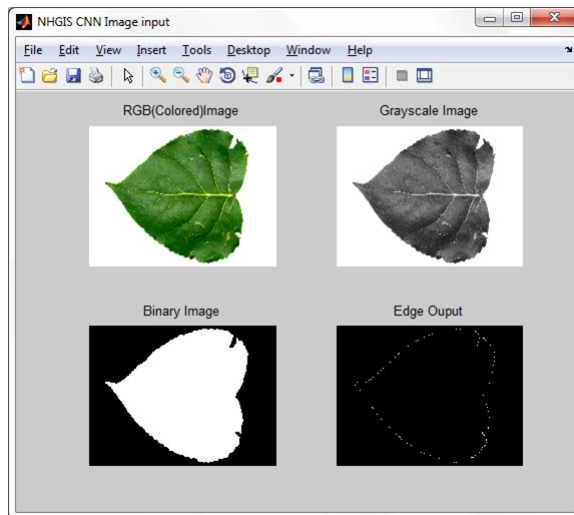


Figure 3: Image pre-processing and segmentation

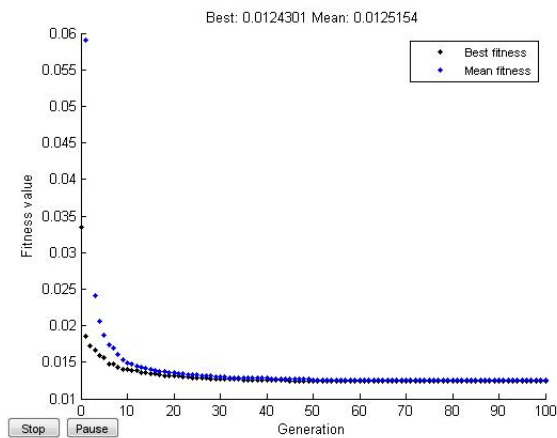


Figure 4: Fitness plot for the GA

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 [7] and [17] respectively. Twenty features were extracted from the Flavia dataset disussed in section . These features are represented in Table 2.

Table 2: 20 features extracted from the Flavia Dataset

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

5.5 Learning system based on Radial Basis Networks

The steps involved in the design of the classification system shown in Figure 6 are described in Figure 5. 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. The parameters of RBF in this work were fixed and pre-determined. Each node in the input layer of the RBF corresponds to a feature vector from Table 2. 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 5.8) which was used in this work. The x in equation 5.8 is the training sample, c_i is the hidden i th 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 [16]. The output value for the training set or any input to the RBF is expressed as equation 5.9. The w in equation 5.9 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.

$$h(||x - c_i||) = \exp \left(-\frac{||x - c_i||^2}{2\sigma^2} \right) \quad (5.8)$$

$$y(x) = \sum_{i=1}^K w_i * h(||x - c_i||) + b \quad (5.9)$$

6 Experimental validation and Results

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

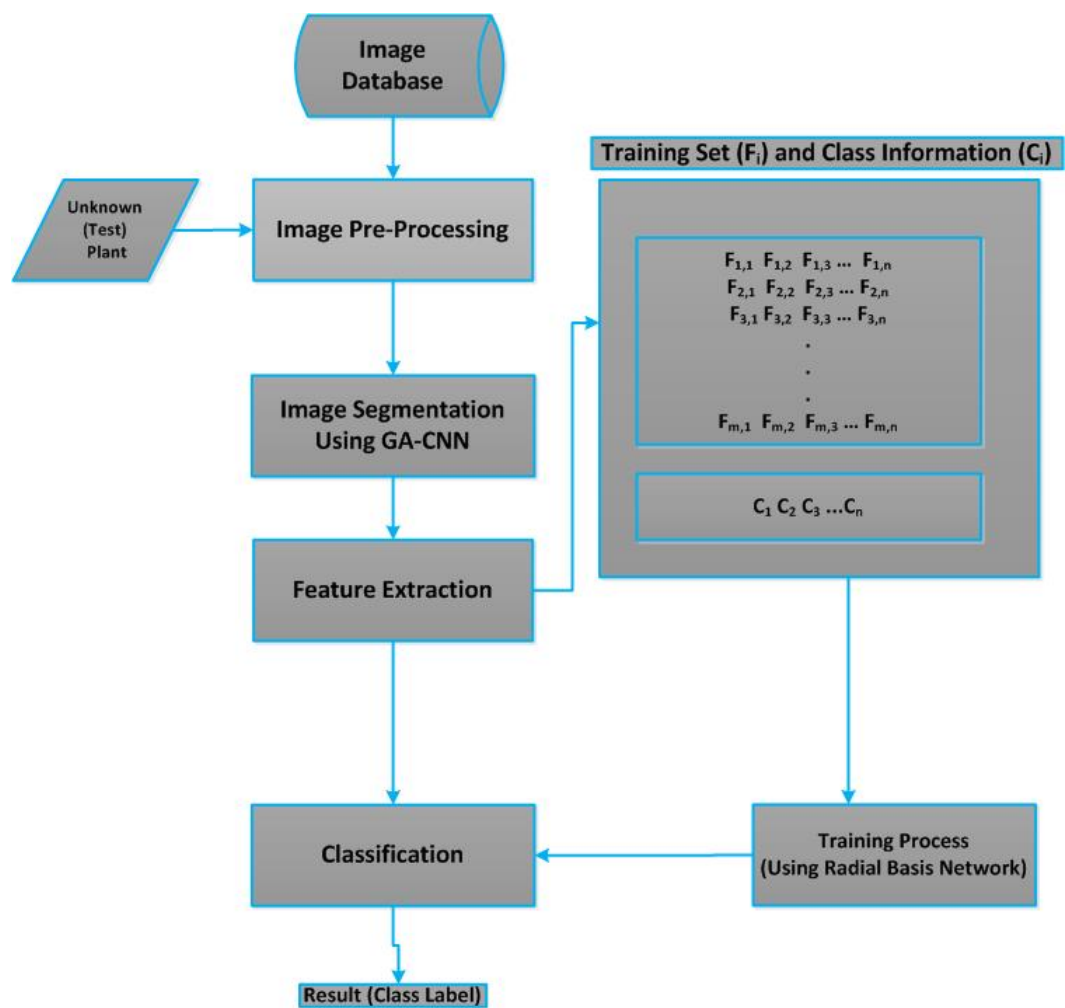


Figure 5: Learning system based on Radial Basis Network

and testing sets as shown in [7]. 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 2 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 $(X, c_i) \in \mathbb{R}^D \times \{1, 2, 3, \dots, 32\}$
- 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 $Error_{RBF}$ 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

$$CVError = \frac{1}{N} \sum_{j=1}^N Error_{RBF}$$

- 9: **end procedure**

7 Results and Discussion

The results of this work are shown in Table 3. 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 (5.5, 5.7, 5.6). In the diagram shown in figure 3, 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 8. The average width of the RBF neurons is 0.45 with best accuracy occurring 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 figure 6.

Table 3: 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

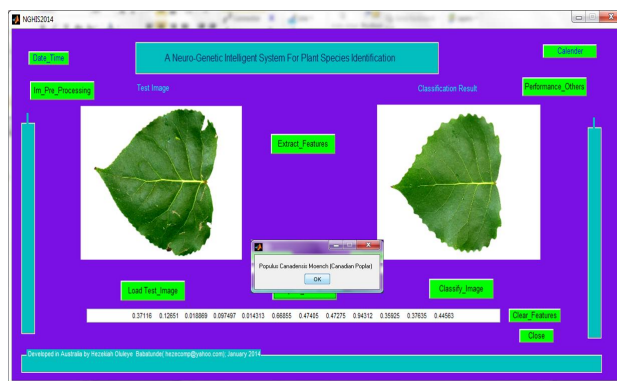


Figure 6: Classification Result using an unknown Plant Specie as Test Image

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 3). This work is adaptive and may further be extended by training the RBF ANN to determine it weights and centres. On way to do this is to employ other optimization techniques apart from GA and make some analysis and comparison.

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 $f(\cdot)$ 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) $\phi(d) = d^n, n = 1, 3, 5, \dots$
 - (b) $\phi(d) = d^n \ln(n), n = 2, 4, 6, \dots$
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 7: Classes of Leaves in the Flavia dataset

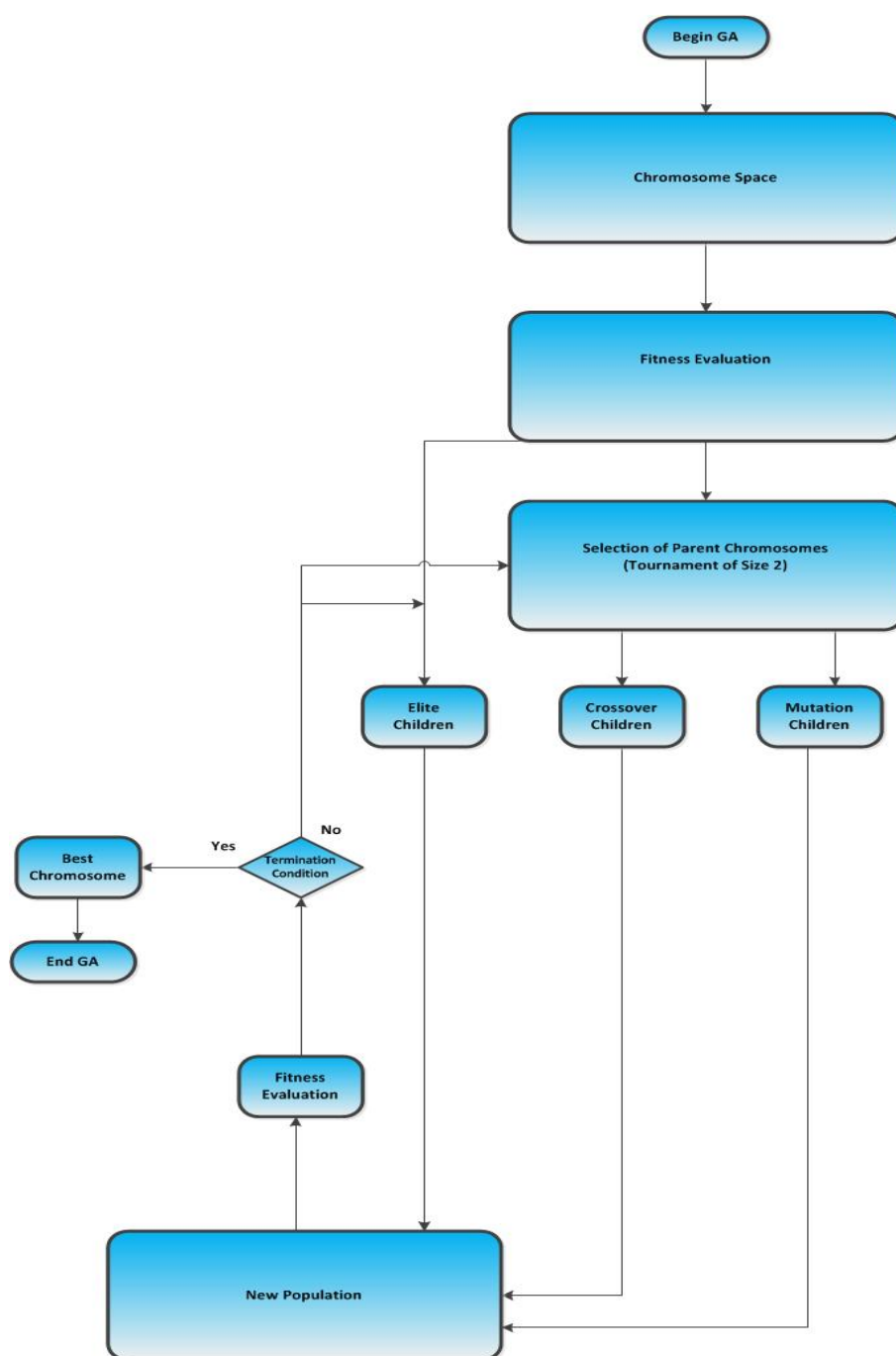


Figure 8: Steps involved in the GA

References

- [1] E H L Aarts and J H M Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, New York, 1989.
- [2] D H Ackley. A learning algorithm for boltzmann machines. *Cognitive Science*, 9:147–169, 1985.
- [3] I Aleksander. *Neural Computing Architecture: The Design of Brain-Like Machines*. North Oxford, London, 1989.
- [4] S I Amari. Mathematical foundations of neurocomputing. *Proc.IEEE*, 78:1443–1463, 1990.
- [5] Akhil Arora, Ankit Gupta, Nitesh Bagmar, Shashwat Mishra, and Arnab Bhattacharya. A plant identification 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*, pages 1–14, 2012.
- [6] H O Babatunde, C O Akanbi, O G Fadare, A A Eludire, O B Aluko, and O G Egbedokun. On numerical simulation of a boundary-valued neuronal model. *World J of Engineering and Pure and Applied Sci, WJEPAS*, 20(2):20–25, 2012.
- [7] Oluleye Babatunde, L. Armstrong, J Leng, and D Diepeveen. Zernike moments and genetic algorithm: Tutorial and application. *British Journal of Mathematics and Computer Science*, 4(15):2217–2236, 2014.
- [8] M Biey, P Checco, and M Gilli, M. Bifurcations and chaos in cellular neural networks. *Journal of Circuits, Systems and Computers*, 12(04):417–433, 2003.
- [9] D S Broomhead and D Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. *Royal Signals and Establishment Malvern (United Kingdom)*, No. RSRE-MEMO-4148, 1988.
- [10] L Changbing and H Wei. Application of genetic algorithm-rbf neural network in water environment risk prediction. *In Computer Engineering and Technology (ICCET). 2nd International Conference on.*, 7:239–242., 2010.
- [11] Pornpanomchai Chomtip, Kuakiatngam Chawin, Supapathranon Pitchayuk, and Siriwisokul Nititit. Leaf and flower recognition system (e-botanist). *IACSIT International Journal of Engineering and Technology*, 3(4):10–15, 2011.
- [12] L O Chua and T Roska. *Cellular neural networks and vision computing*. Cambridge University Press, 2002.
- [13] L O Chua and L Yang. Cellular neural networks: theory and applications. *IEEE Trans on Circuits and System*, 35(10):1257–1272, 1988.
- [14] James S Cope, David Corney, Jonathan Y Clark, and 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*, pages 1–21, 2011.
- [15] K David, P James, and P Mathew. Automatic plant leaf classification for a mobile field guide (an android application). *Department of Electrical Engineering, Stanford California.*, 2012.
- [16] H Demuth, M Beale, and M Hagan. *Neural Network Toolbox Users Guide*. The MathWorks. Inc., Natick, MA, 2013.
- [17] J Fourier, J. B. & Freeman. *The Analytical Theory of Heat*. The University Press, 1878.
- [18] S Haykin. *Redes neurais - principios e prticas*. Bookman, 2001.
- [19] B Hezekiah, A T Akinwale, and O Folorunso. A cellular neural networks- based model for edge detection. *Journal of Information and Computing Science*, 5(1):003–010, 2010.

-
- [20] John H Holland. Genetic algorithms. *Scientific American*, 1962.
 - [21] C Jyotismita and P Ranjan. Plant leaf recognition using shape based features and neural network classifiers. *International Journal of Advanced Computer Science and Applications (IJACSA)*, pages 41–47, 2011.
 - [22] T Kurban and E Besdok. A comparison of rbf neural network training algorithms for inertial sensor based terrain classification. *Sensors*, pages 6312–6329, 2009.
 - [23] K Man, K Tank, and S Kwong. Genetic algorithm: Concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5):519–534, 1996.
 - [24] J L.Orr Mark. Introduction to radial basis function networks. *Recent Advances in Radial Basis Function Networks. Centre for Cognitive Science, University of Edinburgh, Scotland*, pages 1–67, 1996.
 - [25] K Marzuki, R Yusof, S Anis, and K Mohd. 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.*, pages 1702–1708, 2011.
 - [26] K Meeta, K Mrunali, P Shubhada, P Prajakta, and B Neha. Survey on techniques for plant leaf classification. *International Journal of Modern Engineering Research (IJMER)*, 1(2):538–544, 2012.
 - [27] K K Pahalawatta. 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*, pages 1–127, 2008.
 - [28] Arya Sandeep and Lehana Parveen. Development of a seed analyzer using the techniques of computer vision. *International Journal of Distributed and Parallel Systems (IJDPS)*, 3(1):149–155, 2012.
 - [29] Scott E Umbaugh. *Digital Image Processing and Analysis. Human and Computer Vision Applications with CVIPtools*. CRC Press: Taylor & Francis Group, Florida, USA, 2011.
 - [30] Z Wang, Z Chi, and D and Feng. Shape based leaf image retrieval. *IEEE Proceedings*, (online 20030160), 2003.
 - [31] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. *IEEE 7th International Symposium on Signal Processing and Information Technology, Cairo, Egypt; ArXiv 0707.4289 v1 [CS.AI]*, 2007.
 - [32] Xiao, G Ji-Xiang, and W Xiao-Feng. Leaf recognition based on the combination of wavelet transform and gaussian interpolation. *China*, 2005.
 - [33] Z Zalikha, S Puteh, S Itaza, and A Mohtar. Plant identification using moment invariants and general regression neural network. *11th International Conference on Hybrid Intelligent Systems (HIS)*, pages 430–435, 2011.